

**(19) World Intellectual Property Organization
International Bureau**



(43) International Publication Date
21 June 2001 (21.06.2001)

PCT

(10) International Publication Number
WO 01/45007 A1

(51) International Patent Classification⁷: G06F 17/60

(21) International Application Number: PCT/US00/33017

(22) International Filing Date: 6 December 2000 (06.12.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/168,754	6 December 1999 (06.12.1999)	US
60/194,880	6 April 2000 (06.04.2000)	US

(71) Applicant: BIOS GROUP INC. [US/US]; 317 Paseo de Peralta, Santa Fe, NM 87501 (US).

(72) Inventors: **MACREADY, William, G.;** 339 1/2 Delgado, Santa Fe, NM 87501 (US). **EL-BELTAGY, Mohammed;** 2491 Sawmill Road, Apt. 204, Santa Fe, NM 87505 (US). **ROY, Barbeau;** 68 Rosemont Place, San Francisco, CA 94103 (US). **ANDERSON, Mark;** 4129 S. Meadow Road #1824, Santa Fe, NM 87501 (US).

(74) Agents: **MORRIS, Francis, E. et al.**; Pennie & Edmonds LLP, 1155 Avenue of the Americas, New York, NY 10036 (US).

(81) Designated States (national): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LK, LC, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

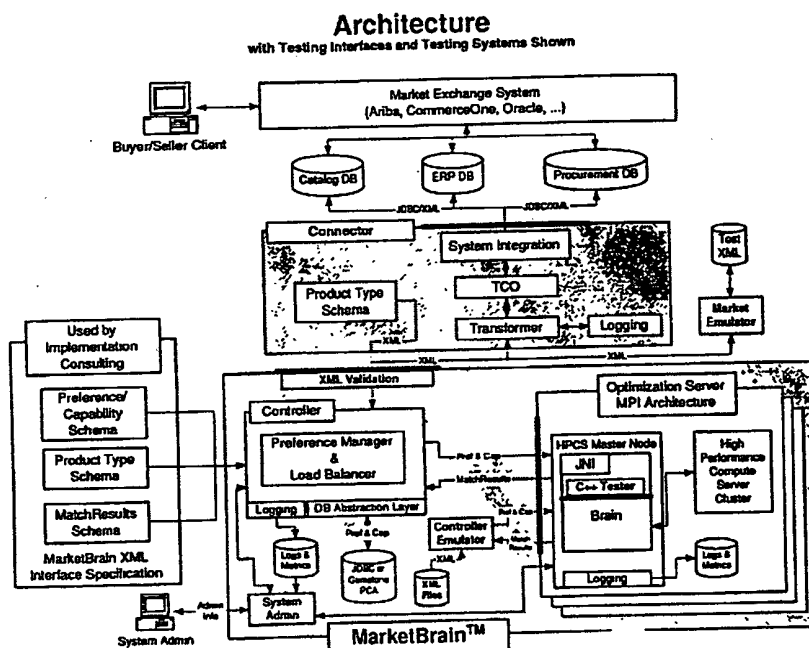
(84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:

— *With international search report.*

[Continued on next page]

(54) Title: A METHOD AND SYSTEM FOR DISCOVERY OF TRADES BETWEEN PARTIES



(57) Abstract: A system is described which allows buyers to define their preferences and sellers to define their capabilities (Fig. 2) and then determines which trading points maximize the utility of the buyer. The system suggests trades by exploiting the flexibilities providing win-win trades. A second level of optimization ranks the trades with all support alternatives. The system allows for rich negotiations.

8.

A Method and System for Discovery of Trades Between Parties

1 Related Applications

This application claims priority to provisional application no. 60/168,754 filed on December 6, 1999, titled, "An E-Commerce Infrastructure for Value Chains", the contents of which are herein incorporated by reference.

2 Field of the Invention

The invention relates to a method and system for discovery of trades between parties. In particular, the invention is a system which allows buyers to define their preferences and sellers to define their capabilities, then determines which trading points maximize the utility of the buyer. The system suggests trades by exploiting the flexibilities and tradeoffs encoded by both parties, thus providing win-win trades. A second level of optimization ranks the trades with all suppliers, allowing the buyer to rapidly determine the best alternatives. The system allows for rich negotiation spaces and supports continuous, discrete, and range or interval decision factors.

3 Background of the Invention

The present invention relates to methods of automatic exploration and exploitation of the flexibilities possessed by negotiating parties to uncover improved win-win agreements. The invention describes computationally efficient mechanisms that are applicable whether there are one or many selling parties. The precise number and types of negotiating dimensions are irrelevant as long as they are numerical. Thus the present invention applies equally to the optimal determination of terms in the purchase of a commodity or an arbitrarily complex artifact.

Electronic markets have proliferated over the last few years with the advent of B2C (business-to-consumer) and B2B (business-to-business) electronic commerce. Such market places have yielded significant cost savings

by lowering the transaction costs between buyers and sellers. Buyers have also profited through increased competition between suppliers. However, electronic markets have hurt suppliers, since the zero-sum negotiation over price has been at their expense. The present invention describes a tool whereby cost savings for both parties are derived from the discovery of win-win trades. Fundamentally, the system works by allowing trading parties to describe their desired trade across multiple dimensions and to express their flexibility around this ideal trade. Through an algorithmic exploration of their flexibilities, the present invention can discover trades that are near the ideal trades of both parties, enabling both to win.

The adoption of B2B and B2C electronic commerce was facilitated by the migration of catalogues online. This familiar method of presentation ameliorated the significant cultural change to electronic trade. For the foreseeable future, electronic commerce will be dominated by online catalogs. At present, online catalogues are direct translations of their hardcopy counterparts where the attributes of a product are described and a price quoted. Inevitably however, online catalogs will become more expressive. Catalog entries will be able to represent price breaks for large quantity orders, lot sizes, etc. Thus it is important that any software (like the present invention) that uncovers mutually beneficial trading scenarios is able to operate with such catalogs. Consequently, in the present invention there is an asymmetry between buyer (usually a human) and seller (usually an online catalog).

One of the reasons catalogs have come to dominate electronic commerce is that the types of goods that can be represented in catalogs are simple. Whether the product is pens or paper clips, different vendor's offers differ little from each other (a pen is a pen is a pen), and a quick scan of a catalog gives a buyer enough information to make an informed purchase. These types of goods are low margin and inexpensive. In contrast, the vast amount of purchasing between businesses involves materials which are directly connected with business operations - car parts, turbines, etc. Such direct goods are the future of electronic commerce. Unlike present-day engines, any truly useful procurement tool must be able to support direct materials with complex attributes and complex inter-relationships between its components.

Electronic commerce offers unprecedented opportunities for more informed decision-making for both buyers and sellers. The past few decades have seen the widespread adoption of enterprise resource planning (ERP) systems, to the point that now almost every major company has some form of ERP software. ERP functions as the digital nervous system of a company, transmitting and logging information between the company's many different business functions. ERP software keeps track of inventory, monitors the state of purchase orders, signals when a company should reorder direct and indirect materials, and a myriad of other functions. Consequently, ERP databases are a rich source of information to optimize a company's operations. Yet today this information is rarely used to make more informed buying and selling decisions. The present invention can utilize such information sources to optimize a company's interactions with suppliers and customers.

One important manner in which this optimization can occur is through an analysis of *all* cost factors. Current buying and selling practices often focus on limited goals, e.g., minimize the total purchase price. Myopic purchasing strategies often result in higher total cost of ownership when all cost factors relevant to a product in its lifetime of use are included. These other cost factors can be significant. Why save the money in taking delivery two days late if the receiving docks will be full at that time and an additional shift needs to be hired to clear the docks? Why order the cheaper drill bit if it is much more expensive to replace when it breaks? The present invention improves trades by minimizing the total cost of ownership of a product, yielding significant savings to its users. Many total cost factors are difficult to quantify – e.g. what is the cost of dealing with a unionized versus a non-unionized supplier? Consequently, the present invention supports qualitative (best guesses and intuition) as well quantitative factors.

All companies are situated in a supply or value chain. At each step in the chain, a company purchases from its suppliers, transforms these inputs, and sells the output to its customers. The termination of the supply chain is the sale of the final product to the end consumer. Since the only influx of external capital comes from the end consumer, companies have realized that they compete not only as individuals but also as entire supply chains. As result, software products have recently become available which attempt to streamline the operations of links within the entire supply chain. This software, variously called supply chain optimization (SCO) or advanced planning and optimization (APO), operates on the basis of forecasted demand at various points within the supply chain. Based on these predictions, plans are generated telling companies how much to produce and how to schedule their operations. SCO systems are a valuable source of intra-company information – data the present invention capitalizes on. Because SCO software relies on forecasted demand, it is only as helpful as the forecast is accurate, and, unfortunately, in many cases demand is very difficult to predict. How can the software know that laundry detergent will go on special at grocery stores in the Northeast in 7 weeks? As a result of the volatility in demand and the many other unpredictable perturbations that plague supply chains, companies keep significant buffers in the form of inventories. In addition to planning, businesses must also be able to adapt to unplanned effects. Such adaptation requires flexibility and a means to exploit that flexibility. The present invention exploits the flexibility of trading parties to streamline the operations of supply chains by smoothing the boundaries between trading parties.

The present invention is therefore a system to allow trading parties to express trading desires and constraints across many and varied different factors. These trading preferences are informed by many different data sources to optimize for a company's internal operations and its connections to its supply chain through an analysis including total cost factors. The flexibility expressed by all trading parties is exploited to locate win-win opportunities for all parties if they exist.

4 Summary of the Invention

We describe the present invention in its application to facilitating trade between buyers and sellers, but note that the mechanisms described are much more general. We can easily imagine, for example, using the present invention to match individuals (with the desires and skills) to projects.

The inspiration for the present invention comes from utility theory developed by economists since the 1960's. Since we are interested in multiple dimensions of negotiation, we draw from the multi-attribute utility theory literature.¹ Utility is an abstract concept which has been formalized in various ways. For the present purposes utility, u , is a number between 0 and 1 representing a party's willingness to trade. Larger values indicate a greater willingness.

4.1 The Negotiation Space

In any negotiation the parties must come to agreement on the factors requiring negotiation. We call these factors dimensions or variables. As an example, when purchasing a car, the buyer may be concerned with price, time of delivery, and color. Each factor price, time, and color is a dimension. Most dimensions can be classed as one of three types: continuous, discrete, or range/interval. A continuous dimension is one like price for which the buyer's utility varies smoothly across that dimension. The buyer's utility at \$23 001.00 is almost the same as the utility at \$23 000. Color is a discrete dimension. Since the car may only be available in black, white, and silver, the domain of this dimension is the finite set of values {black, white, silver}. Moreover, the buyer's utility may be quite different for the three colors. The third class of dimensions is called interval dimensions. An interval dimension arises often in B2B negotiations. If a machined part is built to some tolerance (e.g., the inner diameter of a screw is between 24.5 and 25.5 mm), the range of variability in the dimension is specified as an interval. In the language of statistical quality control, a certain percentage of the machined parts will fall in this range. These three broad classes of variables capture almost all the types of attributes relevant to B2B negotiation.

The present invention operates over any number of continuous, discrete, and range or interval variables. We call the negotiation space X and any point in the negotiation space $(x, \kappa, r) \in X$. It is important to recognize that the single trading point (x, κ, r) may have multiple components, e.g., price = \$23 000, time of delivery = 3 weeks, color = black.

In the present invention, the space of negotiation is agreed upon by all parties involved prior to the commencement of any negotiation. We can, however, imagine more dynamic situations in which dimensions are introduced and discarded over time.

¹For a good introduction to multi-attribute utility theory see [1].

4.2 The Buyer's Utility Function

A party defines its utility function over this space so that every (x, κ, r) is assigned a utility number indicating the party's willingness to trade. We indicate the utility function as $u((x, \kappa, r))$. A great deal of work has been done on the appropriate form for utility functions. In the present invention, we take a simple form for the utility function for two reasons. First, we would like the form of the utility to be conducive to rapid computation. Second we would like the utility to be simple enough to be easily understood by and elicited from users of the invention. With no loss in generality, we write the utility function as $u((x, \kappa, r)) = \exp(-d((x, \kappa, r)))$ where $d(x)$ is interpreted as the distance of trading point (x, κ, r) from the most preferred trade.

So that we can operate against seller catalogs, only the buyer needs to define a utility function. Across the continuous dimensions, the buyer's utility is defined by specifying the most preferred (or ideal) continuous dimensions and the manner in which utility drops off as we move away from this ideal. For the discrete dimensions, the utility is specified in tabular form since there are a finite number of alternatives. Again, the buyer must specify its ideal discrete values and how utility decays away from those values. In section 6.1 we describe how this is accomplished. The range dimensions contribute to utility similarly; the buyer specifies an ideal range and the utility decays for ranges other than the ideal according to their distance from the ideal.

The utility function can also express tradeoffs between variables, e.g., I may take delivery in 5 weeks if the price drops to \$20 000, or I may accept the white car if I can take delivery in 2 weeks. The tradeoffs may be between pairs of continuous dimensions (as in the first case), between pairs of discrete variables, or between continuous and discrete variables (as in the second case).

4.2.1 Normalization and Weighting

When utility is defined over different types of variables, it is important to normalize the contributions of each variable so that the buyer can weight the importance of the various contributions to utility. This is a difficult problem. How should a buyer's color preferences be normalized so that they can be traded off against time of delivery? The present invention solves this problem by requiring that the average distance of any negotiation variable from its ideal value is the same for all dimensions. Since the buyer is more interested in those regions of the negotiation space where the utility is high, the average is weighted by utility. This procedure defines a manner in which to define a baseline where all dimensions contribute equally. Given this baseline, the buyer can then weight the various contributions and obtain useful results.

4.2.2 Utility Elicitation

Since utility is fundamental to the present invention, its elicitation from the buyer is important. Utility may be defined using any of a number of sources:

1. graphical user interfaces associated with the invention
2. standard benchmark criteria applicable to the domain
3. formal methodologies like the analytical hierarchical process [2], or discrete choice analysis [3]
4. inferred through models

We expand briefly upon method 4. As discussed in the background section, it is important to buyers to minimize their total cost of ownership. If we have a function representing these costs as a function of the negotiation variables, and perhaps other factors, this function can be used to infer a utility function which will act to minimize the total costs. Later we describe how this can be accomplished.

4.3 A Supplier's Capabilities

As noted previously, suppliers are treated differently from buyers so that the tool can operate against catalog information with no human intervention required on the part of the seller. In fact, we do not require sellers to define a utility at all.

A supplier cannot offer deals at all points within the negotiation space X , e.g., he certainly can't offer the black car tomorrow for free! A capability then represents the ability of a supplier to deliver and defines a subspace of X . It can include such things as price discounts on large volume orders, variation in delivery time as a function of price, etc. Since these relationships are already specified by businesses in terms of simple rules like *"the price per unit is \$10.00 if 1 to 999 units are ordered and \$9.50 per unit if 1000 or more units are ordered"*, suppliers' capabilities are represented in the present invention by piecewise linear functions.

4.4 Negotiation Constraints

Both parties may have constraints which must be satisfied in order for them to trade. For example, the buyer may not buy the car unless he gets it within 6 weeks, or he may not purchase the car if it is available only in white. These are examples of continuous and discrete constraints, respectively. A continuous constraint sets a requirement on the continuous variables. In the present invention, continuous constraints must be either linear or quadratic. Discrete constraints involve discrete variables. A discrete constraint can be expressed as a list of the allowed (or disallowed) combinations of the discrete variables for which the trade will be acceptable. For example, if the buyer would accept either the black or the silver car, the constraint would

list both these colors as viable. It is important to note that both continuous and discrete constraints may involve one or more variables. We can also express constraints involving both types of variables by allowing the continuous constraints to differ depending on the discrete variables.

4.5 Utility Optimization

With the major components of the invention in place, we describe how the overall system works. As a procurement tool for the buyer, there are two levels of optimization. First, for any given supplier we maximize the buyer's utility, subject to the supplier's capabilities to find that trade which makes the buyer as happy as possible. Since we are optimizing within a supplier's capabilities, the supplier has expressed a willingness to complete the trade at whatever point is determined to be optimal. The tool then optimizes across suppliers to rank them according to utility at the optimal point. A graphical user interface allows a buyer to investigate the trades suggested by the tool by sorting according to any dimension or by the overall utility.

Utility, while a useful concept in assessing an overall score, may be of limited use to a buyer due to its abstract meaning. Consequently, we can also apply the total cost of ownership function to the results to rank order the suggested trades according to their various cost components. Recall that for any trade $x \in X$, the total cost of ownership function returns the various cost contributions. This additional information aids the buyer in his purchasing decision. The utility number for each trade is still useful because the total cost of purchase function includes only those cost factors which can be quantified, whereas the utility also includes "softer" qualitative factors.

4.5.1 Aggregation

In addition to optimizing against one supplier at a time, the present invention can also be used to optimize against an arbitrary aggregation of suppliers. This is important if, for example, no single seller can supply the large volume requested by a buyer. In this mode of operation, the buyer specifies sets of suppliers participating in the aggregation and the dimensions over which aggregation can occur, and the tool finds the optimal combination in which to distribute the volume dimension over the allowed suppliers.

5 Brief Description of the Figures

Figure 1 shows an architecture for the invention.

Figure 2 shows a schematic of a buyer-specific capability with examples indicating potential input.

Figure 3 shows a schematic of a supplier-specific preference with examples indicating potential input.

n_c	number of continuous dimensions
n_d	number of discrete dimensions
n_r	number of range dimensions
\mathbf{x}	n_c -vector of values for continuous dimensions
\mathbf{x}	n_d -vector of values for continuous dimensions
x_i	value of i th continuous variable
x_i	value of i th discrete variable

Table 1: Definition of the negotiation search space.

6 Detailed Description

6.1 Theory

In this section we outline the mathematical foundations of the optimization process in sufficient detail to allow for computer implementation.

6.1.1 The Negotiation Space

In Table 1 we define the parameters which collectively define the space of negotiation X . For each of the n_c continuous variables, we specify an allowed range over which that continuous dimension may vary as $x_i \in X_i = [\underline{x}_i, \bar{x}_i]$, where \underline{x} is the n_c -vector of lower continuous bounds and \bar{x} is the n_c -vector of upper continuous bounds. Each discrete variable assumes a value from within its domain $x_i \in \mathcal{D}_i$. Without loss of generality, we label the domain of discrete variable i by $\mathcal{D}_i = [1, \dots, d_i]$ where $d_i \geq 0$ is an integer giving the number of possible values discrete variable x_i may assume.

With these definitions, we define the space of negotiation by the tensor product $X = X_1 \otimes \dots \otimes X_{n_c} \otimes \mathcal{D}_1 \otimes \dots \otimes \mathcal{D}_{n_d}$. Range variables are treated separately and not negotiated over.

6.1.2 The Utility Function

The utility function is a mapping from X into the interval $[0, 1]$. As indicated earlier we assume the utility to have the form $u(\mathbf{x}, \mathbf{x}) = \exp[-d(\mathbf{x}, \mathbf{x})]$ where $d(\mathbf{x}, \mathbf{x})$ is interpreted as a distance. In what follows we will assume that in its simplest form the distance function has the form

$$d(\mathbf{x}, \mathbf{x}, \mathbf{r}) = (\mathbf{x} - \mu(\mathbf{x}))^t \mathbf{C}^{-1}(\mathbf{x}) (\mathbf{x} - \mu(\mathbf{x})) + Z(\mathbf{x}) + R(\mathbf{r}; \mathbf{x}).$$

Each contribution to the distance function is positive. We consider each contribution to the distance in turn, beginning with the range variable contribution $R(\mathbf{r}; \mathbf{x})$.

First, we note that the range distance depends on the setting of the discrete variables. This allows the buyer to express different preferences for the range variables depending on discrete factors. The total range distance is summed up over all possible range variables so that $R(r; \kappa) = \sum_{i=1}^{n_r} R_i(r_i; \kappa)$. The vector r indicates the preferred values for all range variables. If range variable i is specified as the interval $r_i \equiv (\underline{r}_i, \bar{r}_i)$ (where $\bar{r}_i > \underline{r}_i$) then r is an n_r -vector of such tuples. The distance contribution, R_i , from one range variable will depend on the application. If the range variables are meant to represent the tolerances on machined parts where issues of statistical quality control are important, then the distance between two ranges might be related to the overlap between Gaussian distributions. If r_i is interpreted as a Gaussian having mean $(\underline{r}_i + \bar{r}_i)/2$ and standard deviation proportional to $\bar{r}_i - \underline{r}_i$ then an appropriate range distance is given in Appendix A. Other choices for the range distance function are certainly possible.

The continuous distance is quadratic and determined by the positive semidefinite $n_c \times n_c$ matrix C^{-1} . We have allowed this matrix to vary with the setting of the discrete variables and indicated this explicitly through $C^{-1}(\kappa)$.² The n_c -vector μ may also depend on κ and indicates the point at which the utility is maximal – μ is thus identified with the ideal value for the continuous variables. The precise quadratic form is convenient, but, using recent developments in interior point methods, other convex functions are also computationally tractable [4].

The discrete distance is determined through the function $Z(\kappa)$ which maps the discrete space $\mathcal{D}_1 \otimes \dots \otimes \mathcal{D}_{n_d}$ onto the positive real line $[0, \infty]$. In keeping with the assumption that distance is a function of only pairs of components x_i, x_j , we assume the discrete distance has the form³

$$Z(\kappa) = \sum_{i=1}^{n_d} \left\{ Z_i(x_i) + \sum_{j=1(\neq i)}^{n_d} Z_{i,j}(x_i, x_j) \right\}.$$

Each contribution $Z_{i,j}$ is a table consisting of $d_i d_j$ entries, where $Z_{i,j}(x_i, x_j)$ can be interpreted as the distance if discrete dimension i has value x_i conditioned on discrete dimension j having value x_j . The diagonal terms Z_i offer an unconditional distance. The most preferred value for the i th discrete dimension is that for which $Z_i(x_i) = 0$.

Rather than require the user to enter the distances explicitly, there are numerous ways in which the distances can be generated automatically based upon a buyer's ranking of preferred values. Further details can be found in Appendix B.

Weighting of Dimensions

In many cases it is important for simple modifications of the distance function to re-weight the contributions to the total distance. If w_c is an n_c -vector of weights for the continuous dimensions, we can accomplish this by letting $C_w^{-1} = W_c C^{-1} W_c$ where W_c is the diagonal matrix $W_c = \text{diag}(w_c)$.⁴ In a similar way we modify the

²Where no confusion will arise we eliminate the dependence of C^{-1} on κ for notational simplicity.

³Later we shall generalize this distance to include weighting of dimensions.

⁴ $M = [m_{i,j}] = \text{diag}(v)$ is the diagonal matrix formed by setting $m_{i,i} = v_i$ and $m_{i,j} = 0$ for $j \neq i$.

discrete distance to $Z_{w,i,j}(x_i, x_j) = w_{d,i} w_{d,j} Z_{i,j}(x_i, x_j)$ where w_d is the n_d -vector of weights for the discrete variables and $w_{d,i}$ is its i th component. The range contribution is also modified so that $R_{w,i}(r_i) = w_{r,i} R_i(r_i)$ where w_r is the n_r -vector of weights for the range variables and $w_{r,i}$ is its i th component. For convenience the weights are normalized so that $(1^t w_c)^2 + (1^t w_d)^2 + 1^t w_r = 1$. With little additional complexity the dimension weights can be made dependent on the setting of the discrete variables but we will assume throughout that the weights are constant.

With these modifications, the total distance function becomes

$$d(x, x) = (x - \mu(x))^t C_w^{-1}(x) (x - \mu(x)) + Z_w(x) + R_w(r). \quad (1)$$

where $Z_w(x) = \sum_{i=1}^{n_d} w_{d,i} \{w_{d,i} Z_i(x_i) + \sum_{j=1(\neq i)}^{n_d} w_{d,j} Z_{i,j}(x_i, x_j)\}$ and $R_w(r) = \sum_{i=1}^{n_r} w_{r,i} R_i(r_i)$

Assigning weighting factors is useful only if the relevant contributions have been previously normalized so that they are all roughly the same magnitude. This serves as the baseline for which all weights are equal. The question immediately arises as to what criteria to use to weight the distance contributions.

We shall determine scaling factors, $Q_d > 0$ and $Q_r > 0$, so that the average distances per dimension of the discrete, range, and continuous contributions are equal, where by average we mean a utility-weighted average over the entire space of possible trades. This weighting places more emphasis on the better trades

If d_c , d_d , and d_r are the continuous, discrete, and range contributions to the total distance, then after multiplication by the scaling factors $d = d_c + Q_d d_d + Q_r d_r$. The scaling factors are determined through the utility weighted average distances defined by

$$\begin{aligned} \langle d_r \rangle &\equiv \frac{\sum_x \sum_r \int_V du Q_r d_r \exp[-Q_r d_r - Q_d d_d - d_c]}{\sum_x \sum_r \int_V du \exp[-Q_r d_r - Q_d d_d - d_c]} = Q_r \frac{\sum_x \exp[-Q_d d_d] \sum_r d_r \exp[-Q_r d_r] \int_V du \exp[-d_c]}{\sum_x \exp[-Q_d d_d] \sum_r \exp[-Q_r d_r] \int_V du \exp[-d_c]} \\ \langle d_d \rangle &\equiv \frac{\sum_x \sum_r \int_V du Q_d d_d \exp[-Q_r d_r - Q_d d_d - d_c]}{\sum_x \sum_r \int_V du \exp[-Q_r d_r - Q_d d_d - d_c]} = Q_d \frac{\sum_x d_d \exp[-Q_d d_d] \sum_r \exp[-Q_r d_r] \int_V du \exp[-d_c]}{\sum_x \exp[-Q_d d_d] \sum_r \exp[-Q_r d_r] \int_V du \exp[-d_c]} \\ \langle d_c \rangle &\equiv \frac{\sum_x \sum_r \int_V du d_c \exp[-Q_r d_r - Q_d d_d - d_c]}{\sum_x \sum_r \int_V du \exp[-Q_r d_r - Q_d d_d - d_c]} = \frac{\sum_x \exp[-Q_d d_d] \sum_r \exp[-Q_r d_r] \int_V du d_c \exp[-d_c]}{\sum_x \exp[-Q_d d_d] \sum_r \exp[-Q_r d_r] \int_V du \exp[-d_c]} \end{aligned}$$

A few comments on the above equations are in order. First, \sum_x indicates the repeated sum $\sum_{x_1} \cdots \sum_{x_{n_d}}$ over all possible discrete trades. \sum_r indicates a sum over all the range variables and the integral over volume V indicates integration over the continuous trading volume of interest. Finally, we have not included a scaling factor Q_c on the continuous distance, since this can be made equal to 1 if we reinterpret Q_r as Q_r/Q_c and Q_d as Q_d/Q_c .⁵ Each of the averages is an explicit function Q_d and Q_r .

The requirement on equal average contributions determines the two unknowns Q_r and Q_d through the equations: $\langle d_r \rangle / n_r = \langle d_c \rangle / n_c$ and $\langle d_d \rangle / n_d = \langle d_c \rangle / n_c$. These two nonlinear equations are coupled in terms of Q_r and Q_d and must be solved simultaneously for Q_r and Q_d . Further details are found in Appendix C.

⁵We singled the continuous variables out, since there will always be continuous variables in any trading scenario.

x_1	x_2	x_3
1	2	3
1	3	2
2	1	3
2	3	1
3	1	2
3	2	1

(a)

x_2
2

(b)

Table 2: An example set of constraints involving 3 variables where the domains of all variables are $\mathcal{D}_1 = \mathcal{D}_2 = \mathcal{D}_3 = \{1, 2, 3\}$. Constraint (a) requires that the values assumed by x_1 , x_2 , and x_3 are all different from each other, and constraint (b) requires that the value assumed by x_2 is even. See text for the solution to both constraints.

6.1.3 Constraint Specification

Buyers and sellers may express constraints over both continuous and discrete variables.

Continuous Constraints

For simplicity (and because additional expressiveness is rarely required) we assume that the buyer's constraints over the continuous variables are linear.⁶ This allows a buyer to express a constraint, e.g., the time of delivery must be within 10 days or I will not trade, i.e., $t \leq 10$. We allow for both inequality and equality constraints which can be expressed as $G_1^{(b)}x = g_1^{(b)}$ and $G_2^{(b)}x \leq g_2^{(b)}$. If there are $c_1^{(b)}$ equality constraints then $G_1^{(b)}$ has $c_1^{(b)}$ rows. Similarly, $G_2^{(b)}$ has $c_2^{(b)}$ rows if there are $c_2^{(b)}$ inequality constraints. We allow the constraints to depend on the setting of the discrete variables, and to be explicit we often write $G_1^{(b)}(x)$, $g_1^{(b)}(x)$, $G_2^{(b)}(x)$, and $g_2^{(b)}(x)$.

Discrete Constraints

We use a standard methodology to represent and process constraints over discrete variables [5]. Abstractly, a constraint over a (perhaps proper)⁷ subset of the discrete variables is represented as a list of all the allowed combinations the variables may assume. An example representation of a pair of discrete constraints is given in Table 2. There are two solutions to this set of constraints: $x_1 = 1, x_2 = 2, x_3 = 3$ and $x_1 = 3, x_2 = 2, x_3 = 1$. We indicate these solutions as $\{(x_1, 1), (x_2, 2), (x_3, 3)\}$ and $\{(x_1, 3), (x_2, 2), (x_3, 1)\}$ respectively. Each solution where all the variables have been identified with specific values from their domains is called a labelling.

⁶With little additional complexity we can also handle quadratic constraints, see [4].

⁷A proper subset of a set is the set itself.

Computationally efficient representations are used to ensure that only feasible combinations of variables are ever processed. Numerous third-party libraries offer constraint programming functionality.⁸

6.1.4 Utility and Total Cost of Ownership

The buyer's utility function and associated constraints may be difficult for many users to define. In this section we show how models of the buyer's business can be used to define utility in a natural manner.

We imagine a function which provides an estimate of the total cost of ownership for any given purchase. Cost contributions to this function might include piece part costs, freight costs, setup costs, quality assurance costs, repair costs, etc. It is important to include all quantifiable costs associated with the lifetime of use of the purchased product because it is this function we will be minimizing. Significant savings may be obtained by taking a longer-term view of the purchase. Revenues (negative costs) generated from the purchase are also included in the function so that the function represents some measure of profitability associated with the purchase. We write the total cost of ownership function as $C_o(x, \kappa, r; \beta)$. We explicitly indicate the dependence on the negotiated trade parameters x , κ , and r , as well as other factors β . The other factors might include forecasted demand, current inventory levels, etc. These factors will vary over time, and they can be extracted from the buyer's ERP and supply chain management systems (SCM) in real-time just before the purchase to ensure continuous real-time optimization. See section 6.2.1 for further details.

Minimization of $C_o(x, \kappa, r; \beta)$ defines an ideal trade dependent on current conditions: $x_{opt}(\beta)$, $\kappa_{opt}(\beta)$, $r_{opt}(\beta)$. If desired, these can be used to define $\mu = x_{opt}(\beta)$, $\tau = r_{opt}(\beta)$ and the desired ideal discrete configuration $\kappa_{opt}(\beta)$ (having distance contribution $Z = 0$). Moreover, the tradeoffs between continuous dimensions around this minimum can be obtained through calculation of the Hessian matrix $H = [h_{i,j}]$ where the i, j matrix element is given by

$$h_{i,j} = \frac{\partial^2 C_o(x, \kappa, r; \beta)}{\partial x_i \partial x_j} \bigg|_{x=x_{opt}(\beta), \kappa=\kappa_{opt}(\beta), r=r_{opt}(\beta)}$$

We then identify C^{-1} with H . In this way, little trading flexibility is obtained in directions where total cost of ownership rises rapidly, while significant flexibility is obtained in directions where total cost of ownership increases slowly.

In summary, a total cost of ownership model defines both the most preferred trade parameters and the flexibility possessed around the preferred trade. The model pulls dynamically from real-time data sources to provide the most up-to-date optimization based on total costs of ownership and other important qualitative factors the buyer may wish to describe in the utility function. The same function and its constituent costs may also be used to help analyze proposed trades from suppliers.

⁸See for example www.mozart-oz.org or www.ilog.com.

6.1.5 Supplier Capabilities

As discussed in the introduction, suppliers represent their capabilities through a specification of the subspace of X in which they will trade. A supplier's capabilities must specify the allowed continuous, discrete, and range variables. The allowed range variables are expressed as the pairs (r_j, \bar{r}_j) , one for each range variable. For example, if a supplier produces 25mm inner diameter screws to within a tolerance of 0.5 mm, then the range variable is simply (24.5, 25.5). These are compared with the buyer's ideal range and contribute to the distance function through the $R_w(\kappa)$ function.

Capabilities over continuous and discrete variables are more complex.

Continuous Capabilities

Continuous capabilities are viewed naturally as responses to a buyer's request. Thus we distinguish between a buyer's requested continuous vector $x^{(b)}$ and a seller's response $x^{(s)}$. A vector-valued function, $f(x^{(b)}, x^{(s)}, \kappa)$ returns the response based on the buyer's request and also, perhaps, other previously defined supplier responses. Component f_i of f defines the i th continuous variable, i.e. $x_i^{(s)} = f_i(x^{(b)}, x^{(s)})$.

Currently, suppliers are used to quoting price discounts for large volume orders and these price discounts are expressed as piecewise linear functions. Consequently, we restrict f_i to have the following form (where we distinguish between the functions depending on the buyer and seller variables):

$$x_i^{(s)} = \sum_k f_{i,k}^{(s)}(x_k^{(b)}, \kappa^{(s)}) + f_{i,k}^{(b)}(x_k^{(b)}, \kappa^{(s)}). \quad (2)$$

An example of how this may be used to define a supplier response is the following: We assume three continuous dimensions – price, volume, and time of delivery and indicate these as $[x_1, x_2, x_3] = [p, v, t]$. Then a response may be formed as

$$\begin{aligned} v^{(s)} &= f_{v,v}(v^{(b)}) \\ t^{(s)} &= f_t(v^{(s)}, t^{(b)}) = f_{t,v}(v^{(s)}) \\ p^{(s)} &= f_p(v^{(s)}, t^{(s)}) = f_{p,v}(v^{(s)}) + f_{p,t}(t^{(s)}). \end{aligned}$$

The $f_{v,v}$ function returns the volume a supplier will fulfill as a function of what the buyer asked for. If the supplier can deliver any volume, this will be the identity function. If the supplier delivers only in certain lot sizes, this function may have a staircase shape, etc. The $f_{t,v}$ function indicates the time it will take a supplier to deliver a certain volume. So, for example, if larger shipments require longer transportation, then this dependence is given by this function. Finally, we turn to the price determination. In this example the price depends on the quantity $v^{(s)}$ being shipped and the $f_{p,v}$ might represent price discounts for large volume orders. There is also an incremental price contribution based on the time of delivery. If faster delivery is more expensive, this is reflected in $f_{p,t}$.

For a given setting of the discrete variables, each $f_{i,k}^{(s)}(x_k^{(s)}, \mathbf{x}^{(s)})$ and $f_{i,k}^{(b)}(x_k^{(b)}, \mathbf{x}^{(b)})$ is a one-dimensional piecewise linear function. Consequently, the functions can be specified by listing the breakpoints. If $f_{i,k}^{(s)}(x_k^{(s)}, \mathbf{x}^{(s)})$ has $\kappa_{i,k}^{(s)}$ breakpoints, then we list these as $\{x_k^{(s)}(b_{i,k}^{(s)} = 1), x_k^{(s)}(b_{i,k}^{(s)} = 2), \dots, x_k^{(s)}(b_{i,k}^{(s)} = \kappa_{i,k}^{(s)})\}$ ⁹ and function values at these breakpoints are $\{f_{i,k}^{(s)}(x_k^{(s)}(b_{i,k}^{(s)} = 1)), f_{i,k}^{(s)}(x_k^{(s)}(b_{i,k}^{(s)} = 2)), \dots, f_{i,k}^{(s)}(x_k^{(s)}(b_{i,k}^{(s)} = \kappa_{i,k}^{(s)}))\}$. Similarly, $f_{i,k}^{(b)}$ is a piecewise linear function defined by the $\kappa_{i,k}^{(b)}$ breakpoints $\{x_k^{(b)}(b_{i,k}^{(b)} = 1), x_k^{(b)}(b_{i,k}^{(b)} = 2), \dots, x_k^{(b)}(b_{i,k}^{(b)} = \kappa_{i,k}^{(b)})\}$ and function values at these breakpoints $\{f_{i,k}^{(b)}(x_k^{(b)}(b_{i,k}^{(b)} = 1)), f_{i,k}^{(b)}(x_k^{(b)}(b_{i,k}^{(b)} = 2)), \dots, f_{i,k}^{(b)}(x_k^{(b)}(b_{i,k}^{(b)} = \kappa_{i,k}^{(b)}))\}$. The breakpoints are indexed by the integers $b_{i,k}^{(s)}$ and $b_{i,k}^{(b)}$.

An interval is specified by assigning a value $b_{i,k}^{(s)} \in [1, \kappa_{i,k}^{(s)} - 1]$ and $b_{i,k}^{(b)} \in [1, \kappa_{i,k}^{(b)} - 1]$ so that¹⁰

$$x_k^{(s)}(b_{i,k}^{(s)}) \leq x_k^{(s)} \leq x_k^{(s)}(b_{i,k}^{(s)} + 1) \quad \forall i \quad \text{and} \quad x_k^{(b)}(b_{i,k}^{(b)}) \leq x_k^{(b)} \leq x_k^{(b)}(b_{i,k}^{(b)} + 1) \quad \forall i. \quad (3)$$

Within each interval the functions are linear, so we have

$$x_i^{(s)} = c_i(\mathbf{x}^{(s)}, \{b_{i,k}^{(s)}\}, \{b_{i,k}^{(b)}\}) + \sum_k m_{i,k}^{(s)}(\mathbf{x}^{(s)}, b_{i,k}^{(s)})x_k^{(s)} + m_{i,k}^{(b)}(b_{i,k}^{(b)})x_k^{(b)}$$

where $c_i(\mathbf{v}^{(s)}, \{b_{i,k}^{(s)}\}, \{b_{i,k}^{(b)}\}) = \sum_k c_{i,k}^{(s)}(\mathbf{x}^{(s)}, b_{i,k}^{(s)}) + c_{i,k}^{(b)}(b_{i,k}^{(b)})$. In the above equations, the intercepts and slopes are given explicitly by

$$c_{i,k}^{(s)}(b_{i,k}^{(s)}) = \frac{x_k^{(s)}(b_{i,k}^{(s)} + 1)f_{i,k}^{(s)}(x_k^{(s)}(b_{i,k}^{(s)})) - x_k^{(s)}(b_{i,k}^{(s)})f_{i,k}^{(s)}(x_k^{(s)}(b_{i,k}^{(s)} + 1))}{x_k^{(s)}(b_{i,k}^{(s)} + 1) - x_k^{(s)}(b_{i,k}^{(s)})}$$

and

$$m_{i,k}^{(s)}(b_{i,k}^{(s)}) = \frac{f_{i,k}^{(s)}(x_k^{(s)}(b_{i,k}^{(s)} + 1)) - f_{i,k}^{(s)}(x_k^{(s)}(b_{i,k}^{(s)}))}{x_k^{(s)}(b_{i,k}^{(s)} + 1) - x_k^{(s)}(b_{i,k}^{(s)})}$$

respectively. An analogous result holds for the $c_{i,k}^{(b)}(b_{i,k}^{(b)})$ and $m_{i,k}^{(b)}(b_{i,k}^{(b)})$.

To eliminate any cyclic dependence on $x_i^{(s)}$ we must impose an ordering on $x_i^{(s)}$ so that $x_i^{(s)}$ can only depend on $x_j^{(s)}$ where $j < i$. Consequently, we can write

$$x_i^{(s)} = c_i(\mathbf{x}^{(s)}, \{b_{i,1}^{(s)}, \dots, b_{i,i-1}^{(s)}\}, \{b_{i,k}^{(b)}\}) + \sum_{k < i} m_{i,k}^{(s)}(\mathbf{x}^{(s)}, b_{i,k}^{(s)})x_k^{(s)} + \sum_k m_{i,k}^{(b)}(b_{i,k}^{(b)})x_k^{(b)}.$$

Written as a matrix equation, the above becomes

$$(\mathbf{I} - \mathbf{M}^{(s)}(\mathbf{x}^{(s)}, \{b_{i,k}^{(s)}\}))\mathbf{x}^{(s)} = \mathbf{c}(\mathbf{x}^{(s)}, \{b_{i,k}^{(s)}\}, \{b_{i,k}^{(b)}\}) + \mathbf{M}^{(b)}(\{b_{i,k}^{(b)}\})\mathbf{x}^{(b)}$$

where $\mathbf{c}(\mathbf{x}^{(s)}, \{b_{i,k}^{(s)}\}, \{b_{i,k}^{(b)}\}) = [c_1(\mathbf{x}^{(s)}, \{b_{i,k}^{(s)}\}, \{b_{i,k}^{(b)}\}) \quad \dots \quad c_n(\mathbf{x}^{(s)}, \{b_{i,k}^{(s)}\}, \{b_{i,k}^{(b)}\})]^t$, $\mathbf{x}^{(s)} = [x_1^{(s)} \quad \dots \quad x_{n_s}^{(s)}]^t$,

⁹The breakpoints are assumed to be in increasing order.

¹⁰Note that many choices for $\{b_{i,k}^{(s)}\}$ or $\{b_{i,k}^{(b)}\}$ would be inconsistent with these constraints. A simple way to fix this is to define a union of all breakpoints (s) and (b) , and have all $f_{i,k}$ have the same breakpoints.

$$\mathbf{x}^{(b)} = [x_1^{(b)} \quad \dots \quad x_{n_c}^{(b)}]^t, \text{ and}$$

$$M^{(s)}(\mathbf{x}^{(s)}, \{b_{i,k}^{(s)}\}) = \begin{bmatrix} 0 & 0 & \dots & 0 & 0 \\ m_{2,1}^{(s)}(\mathbf{x}^{(s)}, b_{2,1}^{(s)}) & 0 & \dots & 0 & 0 \\ m_{3,1}^{(s)}(\mathbf{x}^{(s)}, b_{3,1}^{(s)}) & m_{3,2}^{(s)}(\mathbf{x}^{(s)}, b_{3,2}^{(s)}) & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ m_{n,1}^{(s)}(\mathbf{x}^{(s)}, b_{n,1}^{(s)}) & m_{n,2}^{(s)}(\mathbf{x}^{(s)}, b_{n,2}^{(s)}) & \dots & m_{n,n-1}^{(s)}(\mathbf{x}^{(s)}, b_{n,n-1}^{(s)}) & 0 \end{bmatrix}$$

and

$$M^{(b)}(\{b_{i,k}^{(b)}\}) = \begin{bmatrix} m_{1,1}^{(b)}(b_{1,1}^{(b)}) & m_{1,2}^{(b)}(b_{1,2}^{(b)}) & \dots & m_{1,n}^{(b)}(b_{1,n}^{(b)}) \\ m_{2,1}^{(b)}(b_{2,1}^{(b)}) & m_{2,2}^{(b)}(b_{2,2}^{(b)}) & \dots & m_{2,n}^{(b)}(b_{2,n}^{(b)}) \\ \vdots & \vdots & \ddots & \vdots \\ m_{n,1}^{(b)}(b_{n,1}^{(b)}) & m_{n,2}^{(b)}(b_{n,2}^{(b)}) & \dots & m_{n,n}^{(b)}(b_{n,n}^{(b)}) \end{bmatrix}.$$

In most cases $\mathbf{x}^{(s)}$ will depend only on a subset of the variables in $\mathbf{x}^{(b)}$. If $\mathbf{x}^{(s)}$ depends on $n' \leq n$ of the $\mathbf{x}^{(b)}$ variables, then $M^{(b)}$ is an $n \times n'$ matrix. In the example given everything depending only upon the volume the buyer requested.

Since $M^{(s)}(\mathbf{x}^{(s)})$ is lower triangular and can be inverted in time $O(n)$, we can rapidly express $\mathbf{x}^{(s)}$ as

$$\mathbf{x}^{(s)} = (\mathbf{I} - M^{(s)}(\mathbf{x}^{(s)}, \{b_{i,k}^{(s)}\}))^{-1} \mathbf{c}(\mathbf{x}^{(s)}, \{b_{i,k}^{(s)}\}, \{b_{i,k}^{(b)}\}) + (\mathbf{I} - M^{(s)}(\mathbf{x}^{(s)}, \{b_{i,k}^{(s)}\}))^{-1} M^{(b)}(\{b_{i,k}^{(b)}\}) \mathbf{x}^{(b)} \quad (4)$$

as long as the $b_{i,k}^{(s)}$ are chosen to also satisfy $x_k^{(s)}(b_{i,k}^{(s)}) \leq x_k^{(s)} \leq x_k^{(s)}(b_{i,k}^{(s)} + 1)$. These constraints will be used in section 6.1.6 which formulates the optimization problem.

We also allow a supplier to express additional linear constraints so that, for example, he may represent that he does not deliver on Sunday. Thus the supplier may define the matrices $G_1^{(s)}(\mathbf{x})$, $G_2^{(s)}(\mathbf{x})$, and the vectors $g_1^{(s)}(\mathbf{x})$, $g_2^{(s)}(\mathbf{x})$ such that $G_1^{(s)} \mathbf{x}^{(s)} = g_1^{(s)}$ and $G_2^{(s)} \mathbf{x}^{(s)} \mathbf{x} \leq g_2^{(s)}$. $G_1^{(s)}(\mathbf{x})$ and $G_2^{(s)}(\mathbf{x})$ have $c_1^{(s)}$ and $c_2^{(s)}$ rows respectively.

Discrete Capabilities

It is easy to imagine that a supplier's response on a discrete dimension is highly constrained by the values of the response on other dimensions, e.g., certain product characteristics come only in certain colors and package sizes. Consequently, it is not suitable to explicitly define a response but only to make available a supplier's constraints amongst the discrete variables. Consider 3 discrete dimensions where $x_1 \in \mathcal{D}_1 = [a, b, c]$, $x_2 \in \mathcal{D}_2 = [A, B, C, D]$, and $x_3 \in \mathcal{D}_3 = [\alpha, \beta, \gamma, \delta]$, and assume the supplier has the following 3 constraints

$$C_1(x_1, x_3) = \{(a, \alpha), (a, \beta), (b, \beta), (c, \beta)\},$$

$$C_2(x_2, x_3) = \{(A, \beta), (B, \gamma), (D, \beta)\},$$

$$C_3(x_1) = \{b, c\}.$$

We first note that there are 4 feasible solutions (or product configurations the supplier can meet): $[x_1, x_2, x_3] = [b, A, \beta]$, $[b, D, \beta]$, $[c, A, \beta]$, or $[c, D, \beta]$. Feasible solutions to the constraints define the response $x^{(s)}$ for the discrete variables.

We indicate a supplier's or buyer's collective set of discrete constraints by $C^{(s)}(x)$ and $C^{(b)}(x)$ respectively.

6.1.6 The Optimization Problem

Having defined the necessary components, we now define the optimization task which determines the continuous x^* and discrete x^* parameters of the trade.

Since the trade must be acceptable to the supplier, we maximize the buyer's utility over a supplier's capabilities. Equivalently, we minimize the distance from the buyer's ideal values as

$$[x^*, x^*] = \arg \min_{x^{(s)}, x^{(b)}} \left\{ (x^{(s)} - \mu(x^{(s)}))^t C_w^{-1}(x^{(s)}) (x^{(s)} - \mu(x^{(s)})) + Q_d Z_w(x^{(s)}) + Q_r R_w(w) \right\}$$

where

$$x^{(s)} = (I - M^{(s)}(x^{(s)}))^{-1} c(x^{(s)}) + (I - M^{(s)}(x^{(s)}))^{-1} M^{(b)} x^{(b)}$$

subject to the constraints over continuous variables

$$G_1(x^{(s)})x^{(s)} = g_1(x^{(s)}), \quad G_2(x^{(s)})x^{(s)} \leq g_2(x^{(s)})$$

and the constraints over the discrete variables $C^{(b)}(v^{(s)})$, $C^{(s)}(v^{(s)})$. In the above, we have defined the $(c_1^{(s)} + c_1^{(b)}) \times n_c$ and $(c_2^{(s)} + c_2^{(b)}) \times n_c$ matrices $G_1(x^{(s)})$ and $G_2(x^{(s)})$ by

$$G_1(x^{(s)}) = \begin{bmatrix} G_1^{(s)}(x^{(s)}) \\ G_1^{(b)}(x^{(s)}) \end{bmatrix}, \quad \text{and} \quad G_2(x^{(s)}) = \begin{bmatrix} G_2^{(s)}(x^{(s)}) \\ G_2^{(b)}(x^{(s)}) \end{bmatrix}.$$

The $(c_1^{(s)} + c_1^{(b)})$ - and $(c_2^{(s)} + c_2^{(b)})$ -vectors $g_1(x^{(s)})$ and $g_2(x^{(s)})$ are defined by

$$g_1(x^{(s)}) = \begin{bmatrix} g_1^{(s)}(x^{(s)}) \\ g_1^{(b)}(x^{(s)}) \end{bmatrix}, \quad \text{and} \quad g_2(x^{(s)}) = \begin{bmatrix} g_2^{(s)}(x^{(s)}) \\ g_2^{(b)}(x^{(s)}) \end{bmatrix}.$$

The optimization is accomplished by iterating two distinct phases. Phase one sets the continuous parameters optimally for a given setting of the discrete variables. We define the functions

$$d_1(x, x) = (x - \mu(x))^t C_w^{-1}(x) (x - \mu(x)) + R(r; x) \quad \text{and} \quad d_2(x) = Q_d Z_w(x^{(s)}),$$

The first phase of the optimization is the continuous problem:¹¹

$$x^*(x) = \arg \min_{x^{(s)}} d_1(x^{(s)}, x) \quad \text{subject to} \quad G_1(x)x = g_1(x) \quad \text{and} \quad G_2(x)x \leq g_2(x). \quad (5)$$

¹¹No optimization is required over range variables, since these are specified up front by both buyer and seller and merely add to the total distance.

A detailed discussion on the solution of the phase 1 optimization problem is found in appendix D. The second phase determines the best value of the discrete variables by minimizing over a function of \mathbf{x} alone

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} d_1(\mathbf{x}(\mathbf{x}), \mathbf{x}) + d_2(\mathbf{x}) \quad \text{subject to} \quad C^{(b)}(\mathbf{x}) \wedge C^{(s)}(\mathbf{x}). \quad (6)$$

Further details on the phase 2 optimization are given in Appendix E. Once \mathbf{x}^* has been determined, we find \mathbf{x}^* as $\mathbf{x}^* = \mathbf{x}(\mathbf{x}^*)$.

6.1.7 Aggregation

Often a buyer may be willing to divide an order between multiple suppliers in order to aggregate the required demand or to obtain better deals. In this section, we detail how the present invention supports this aggregate optimization.

Aggregation can only occur over the continuous variables where values may be subdivided. Each continuous variable x_i must be parcelled out amongst a set of suppliers. Consequently, we extend our notation to $x_i \rightarrow \tilde{x}_{i,k}$ giving the contribution of the k th supplier to continuous dimension i . The k th supplier may come from a (perhaps proper) subset of all suppliers. We indicate the set of potentially contributing suppliers as \mathcal{K} and the number of potentially contributing suppliers as $|\mathcal{K}|$.¹²

We restrict the discrete variables to be the same across all potentially aggregated suppliers, i.e., we do not generalize $x_i \rightarrow x_{i,k}$. This simplifying assumption is made for two reasons. First, the size of the discrete optimization problem is smaller and so optimization be performed faster. Second, it may be difficult to elicit from the buyer the allowed discrete alternatives for each supplier. Nevertheless, this generalization is straightforward should the need arise.

This simplifying assumption requires that the union of discrete supplier constraints $C_{\mathcal{K}}(\mathbf{x}) \equiv \bigwedge_{k \in \mathcal{K}} C_k^{(s)}(\mathbf{x})$ yields a feasible solution when combined with the buyer's discrete constraints $C^{(b)}(\mathbf{x})$. A necessary (but not sufficient) condition for satisfaction is then that each constraint satisfaction problem k having constraints $C^{(b)}(\mathbf{x}) \wedge C_k^{(s)}(\mathbf{x})$ has a feasible solution.¹³ Henceforth, we will assume that the set of suppliers, \mathcal{K} , satisfies this condition. If not, those suppliers violating the constraints $C^{(b)}(\mathbf{x}) \wedge C_k^{(s)}(\mathbf{x})$ are eliminated from consideration in \mathcal{K} .

Discrete Search

We must search over the subsets of \mathcal{K} for feasible solutions, which is a combinatorial problem. Fortunately, given a complete labelling of variables, determining the largest subset is easy. For any given labelling of all discrete variables, if each $C^{(b)} \wedge C_k^{(s)} \forall k \in \kappa \subset \mathcal{K}$ is satisfiable, then the union $C^{(b)} \wedge C_{\kappa}^{(s)}$ where $C_{\kappa}^{(s)} = \bigwedge_{k \in \kappa} C_k^{(s)}$ is also satisfiable under the same labelling. The largest subset of variables is found by adding all k which

¹²All work to this point is thus seen as the special case $|\mathcal{K}| = 1$.

¹³It is easily seen this requirement is not sufficient by having $C^{(b)} = \{(\mathbf{x}_1, 1), (\mathbf{x}_1, 2)\}$, $C_1^{(s)} = \{(\mathbf{x}_1, 1)\}$, and $C_2^{(s)} = \{(\mathbf{x}_1, 2)\}$.

have feasible solutions with the buyer. We needn't worry about smaller subsets because the continuous optimization will assign zero values to those if appropriate. Consequently, for any given labelling κ we let $\mathcal{K}(\kappa)$ represent the maximal subset of suppliers for which $\mathcal{C}^{(b)}(\kappa) \wedge \mathcal{C}_K(\kappa)$ is satisfiable. It is this set of suppliers which enter into the continuous optimization. The number of participating suppliers is denoted by $|\mathcal{K}(\kappa)|$.

Continuous Optimization

Optimization over the continuous variables is carried for each labelling κ . Generally speaking, the buyer's utility will not be an explicit function of $x_{i,k}$ but only of x_i . We assume a linear relationship between these two quantities so that¹⁴

$$\mathbf{x} = \Xi \bar{\mathbf{x}}.$$

The $n_c |\mathcal{K}(\kappa)|$ vector $\bar{\mathbf{x}}$ is defined as $\bar{\mathbf{x}}^t = [\bar{x}_1, \dots, \bar{x}_{n_c}]$ where $\bar{x}_i^t = [\bar{x}_{i,1}, \dots, \bar{x}_{i,|\mathcal{K}(\kappa)|}]$. The $n_c \times n_c |\mathcal{K}(\kappa)|$ matrix Ξ is assumed known and typically has the form¹⁵

$$\Xi = \begin{bmatrix} \xi_1^t & 0^t & \dots & 0^t \\ 0^t & \xi_2^t & \dots & 0^t \\ \vdots & \vdots & \ddots & \vdots \\ 0^t & 0^t & \dots & \xi_{n_c}^t \end{bmatrix}$$

where 0 is the K -vector of all zeros and ξ_i is the linear combination relating x_i to the $\bar{x}_{i,k}$. Under our assumptions for Ξ , $x_i = \xi_i^t \bar{\mathbf{x}}_i$. In cases where the buyer wants to accumulate the results from suppliers (e.g., aggregating quantities) $\xi = 1$ is the $|\mathcal{K}(\kappa)|$ -vector of all 1s. In other cases the buyer may take $\xi = 1/|\mathcal{K}(\kappa)|$ so that the time of delivery becomes the average time of delivery across the suppliers. Constraints on \mathbf{x} become constraints on $\bar{\mathbf{x}}$ via

$$\bar{\mathbf{G}}_1(\kappa) \bar{\mathbf{x}} = \mathbf{g}_1(\kappa) \quad \text{and} \quad \bar{\mathbf{G}}_2(\kappa) \bar{\mathbf{x}} \leq \mathbf{g}_2(\kappa) \quad (7)$$

where $\bar{\mathbf{G}}_1(\kappa) = \bar{\mathbf{G}}_1(\kappa) \Xi$ and $\bar{\mathbf{G}}_2(\kappa) = \bar{\mathbf{G}}_2(\kappa) \Xi$. We might also expect the buyer to add additional linear constraints, such as requiring the latest shipment from any supplier to arrive earlier than a certain date, or requiring all deliveries to arrive the same day. There can also be constraints specific to particular suppliers, e.g., the buyer doesn't want any more than 100 units from supplier 5. These can be handled simply as constraints on the individual $\bar{x}_{i,k}$ and added as extra rows to $\bar{\mathbf{G}}_1(\kappa)$, $\bar{\mathbf{G}}_2(\kappa)$, $\bar{\mathbf{g}}_1(\kappa)$, and $\bar{\mathbf{g}}_2(\kappa)$.

With aggregation, the quadratic form to be minimized is $(\Xi \bar{\mathbf{x}} - \boldsymbol{\mu}(\kappa))^t \mathbf{C}_w^{-1}(\kappa) (\Xi \bar{\mathbf{x}} - \boldsymbol{\mu}(\kappa))$ subject to the constraints given in Eq. (7). This minimization can be carried out through a straightforward generalization of the method given in Appendix D.

¹⁴We can also relate \mathbf{x} and $\bar{\mathbf{x}}$ by $\mathbf{x} = \Xi \bar{\mathbf{x}} + \boldsymbol{\nu}$ for some constant vector $\boldsymbol{\nu}$, which will not cause any complications. However, there seems to be little practical reason to do so.

¹⁵With no additional computational complexity, we can allow Ξ to depend on κ .

6.2 Implementation

In this section we outline an implementation of the entire e-procurement invention. We begin with a high-level description of the architecture, then fill in the details by describing a complete object model.

6.2.1 High-level Architecture of the Invention

There are at least two modes in which the invention may be used. First, the invention may reside at the site of large buyers, and suppliers who wish to sell to the buyer may be required to submit their capabilities via a web interface to the buyer. The invention may also be used within a marketplace hosted by a third party. Buyers/sellers log onto the market, submit their preference/capabilities, and act on the results. The architecture is modular enough to support both modes of operation.

In Figure 1 we present an architecture for the invention. We describe the architecture, starting from the optimization algorithm which finds matches between buyers and sellers and work our way outwards.

A controller surrounds the optimization engine, feeding it buyer preferences and seller capabilities. If multiple optimization processes are running (perhaps on different machines), the controller can also do load balancing, forwarding the request to the least busy process. The controller decomposes preferences and capabilities into their constituent buyer- and seller-specific versions (see below), selects the most specific matching preference/capability pairs, and sends them to the matching engine for optimization. The controller then collects responses from the matching engine and returns them to the buyer. Additionally, the controller logs all results into a database for recording purposes.

Another layer, called the Connector in Figure 1, separates the graphical user interface (GUI) through which users communicate with the tool from the controller. This layer serves a number of functions. The connector transforms the description of preferences and capabilities from the GUI into a form suitable for the implementation of the matching engine. Part of this transformation involves validation of appropriate input from the GUI layer so that no malformed input is ever sent to the controller. The Connector layer can also pull data from ERP or SCM systems and automatically infer preferences (using the total cost of ownership function) for the buyer. The enterprise abstraction layer insulates the Connector from the precise details of the manner in which the ERP and SCM data needs to be gathered. Total cost of ownership is evaluated in the simulation modules, which may either be running locally at the client's site or running centrally at the main server. These simulation modules pull operational data (the vector β) from the enterprise abstraction layer. A preference optimization module (TCO) minimizes the total cost of ownership to determine the ideal trade and the flexibilities around the ideal trade.

At the outmost level, a layer provides integration with the GUI and/or host system. A number of administrative systems are expected at this layer. Market administration services allow easy definition of trading spaces, the dimensions of negotiation, limits on continuous variables, allowed settings of the

discrete variables, etc. User administration services allow an administrator to define buyers, passwords, spending limits, etc. Supplier services accomplish similar tasks on the supply side. Managers for preferences, capabilities, and match results ensure that these objects are properly stored in a database. This layer also dynamically generates the html necessary for presentation of the data via a web interface to buyers and sellers.

For maximal portability, communications between the View and Connector are via XML documents. For maximal efficiency, communications between the Connector and matching controller are as serialized Java objects.

6.2.2 An Object Model for the Invention

The fundamental objects required for the invention are preferences from buyers, capabilities from sellers, and match results returned to all parties. The components of such objects have already been considered from a mathematical point of view, and we now describe one possible computer representation.

In this section we describe a complete grammar for the object model. The following syntactic conventions are used:

- $\langle nt \rangle$ denotes a non-terminal symbol nt
- $[obj]$ denotes an optional grammar segment obj
- $\{obj\}$ denotes 1, or many times the grammar segment obj
- \rightarrow denotes a production rule for non-terminal symbol. If there are multiple rules, say $\langle a \rangle$, $\langle b \rangle$, and $\langle c \rangle$, then these are denoted as

$$\langle nt \rangle \rightarrow \langle a \rangle | \langle b \rangle | \langle c \rangle.$$

In contrast, a production rule of the form

$$\langle nt \rangle \rightarrow \langle a \rangle, \langle b \rangle, \langle c \rangle$$

indicates that the non-terminal $\langle nt \rangle$ is composed of three grammar segments, $\langle a \rangle$, $\langle b \rangle$, and $\langle c \rangle$

- terminal keywords are in serif font

Obvious non-terminal grammar elements like $\langle string \rangle$ and $\langle integer \rangle$ are not described.

Supply Side

To represent capabilities that apply to a specific buyer (perhaps for contractual reasons), we have defined a capability to be a list of $\langle buyerSpecificCapability \rangle$. With one exception, a buyer-specific capability applies only to one buyer – that buyer associated in the id field of the $\langle buyerSpecificCapability \rangle$. The exception

occurs if the id field is * or wildcard. This indicates that the capability applies to all buyers. Using buyer-specific capabilities, suppliers can represent specific capabilities to certain buyers and generic capabilities applying to all other buyers. By not including a wildcard (buyerSpecificCapability) and only listing (buyerSpecificCapability)s applicable to specific buyers, sellers can also represent the fact that they will trade only with a subset of all buyers. In cases where both the wildcard (buyerSpecificCapability) and a (buyerSpecificCapability) applicable to a specific buyer apply, the most specific (buyerSpecificCapability) is selected.

A schematic of a (sellerSpecificPreference) is given in Figure 2.

We begin at the top level of a capability:

$$\text{capability} \rightarrow \{(\text{buyerSpecificCapability})\}$$

where

$$\begin{aligned} (\text{buyerSpecificCapability}) \rightarrow & \text{id: } \langle \text{id} \rangle, \\ & \text{discrete: } \{(\text{discreteVarDescription})\}, \\ & \text{continuous: } \{(\text{continuousVarDescription})\}, \\ & \text{range: } \{(\text{rangeVarDescription})\}, \\ & [\text{discreteConstraint: } \langle \text{discreteConstraint} \rangle], \\ & \text{instance: } \{(\text{discreteCapabilityInstance})\} \\ & [\text{aggregationParticipation: } 0|1]. \end{aligned}$$

(id) identifies a buyer or group of buyers. Individual buyers are represented by some unique identifier (say an integer) and the group of all buyers is identified by the wildcard character '*'. So we have

$$\langle \text{id} \rangle \rightarrow \langle \text{integer} \rangle | *.$$

aggregationParticipation is a Boolean flag giving the supplier's willingness to participate in aggregate orders to the identified buyer.

Each of the variable constituent components is described by

$$\begin{aligned} (\text{discreteVarDescription}) \rightarrow & \text{name: } \langle \text{integer} \rangle, \\ & \text{allowedValues: } \{(\text{integer})\} \\ (\text{continuousVarDescription}) \rightarrow & \text{name: } \langle \text{integer} \rangle, \\ & \text{min: } \langle \text{double} \rangle, \\ & \text{max: } \langle \text{double} \rangle \\ (\text{rangeVarDescription}) \rightarrow & \text{name: } \langle \text{integer} \rangle. \end{aligned}$$

In its simplest form, a $\langle \text{discreteConstraint} \rangle$ is a list of more primitive constraints

$$\langle \text{discreteConstraint} \rangle \rightarrow \{ \langle \text{primitiveDiscreteConstraint} \rangle \}$$

where each primitive constraint is composed as follows:

$$\begin{aligned} \langle \text{primitiveDiscreteConstraint} \rangle \rightarrow & \text{name: } \langle \text{string} \rangle \\ & \text{variables: } \{ \langle \text{discreteVarName} \rangle \}, \\ & \text{includes: } 0 \mid 1, \\ & \text{values: } \langle \text{integerMatrix} \rangle \end{aligned}$$

$\langle \text{discreteVarName} \rangle$ is the name of the discrete variable involved in the constraint

$$\langle \text{discreteVarName} \rangle \rightarrow \langle \text{integer} \rangle.$$

The includes field is a bit. If the bit is 1, then the combinations listed in the values field are the allowed values the variables may take on. If the bit is 0, then the combinations listed in values are the excluded combinations, i.e., everything in the powerset of the variables is allowed except those combinations listed in values. The order of the variable names is significant, since they will be assumed to be in the same order in values. If there are a variables involved in the constraint, and c constraints, then $\langle \text{integerMatrix} \rangle$ is an $a \times c$ matrix of integers:

$$\begin{aligned} \langle \text{integerMatrix} \rangle & \rightarrow \langle \text{integerVector} \rangle, \dots, \langle \text{integerVector} \rangle \\ \langle \text{integerVector} \rangle & \rightarrow \langle \text{integer} \rangle, \dots, \langle \text{integer} \rangle \end{aligned}$$

The $\langle \text{discreteCapabilityInstance} \rangle$ component is described by

$$\begin{aligned} \langle \text{discreteCapabilityInstance} \rangle \rightarrow & \text{mask: } \langle \text{discreteVarMask} \rangle, \\ & [\text{rangeCapability: } \{ \langle \text{rangeVarInstance} \rangle \}], \\ & \text{continuousCapability: } \langle \text{continuousCapability} \rangle \\ & \text{continuousConstraints: } \langle \text{continuousConstraints} \rangle \end{aligned}$$

A $\langle \text{rangeVarInstance} \rangle$ defines a range variable and has the form

$$\begin{aligned} \langle \text{rangeVarInstance} \rangle \rightarrow & \text{name: } \langle \text{integer} \rangle, \\ & \text{min: } \langle \text{double} \rangle, \\ & \text{max: } \langle \text{double} \rangle. \end{aligned}$$

The $\langle \text{discreteVarMask} \rangle$ relates to the discussion of 6.2.2. As in Table 3 we have

$$\langle \text{discreteVarMask} \rangle \rightarrow \{ \langle \text{extendedVarValue} \rangle \}$$

where an $\langle \text{extendedVarValue} \rangle$ is either an integer from the domain of the discrete variable or the wildcard character '*':

$$\langle \text{extendedVarValue} \rangle \rightarrow \langle \text{integer} \rangle | *.$$

$\langle \text{continuousConstraints} \rangle$ describes the hard linear constraints for the continuous variables. Since these constraints may be either inequality or equality, we have

$$\begin{aligned} \langle \text{continuousConstraints} \rangle \rightarrow & [\text{equality: } \langle \text{linearConstraints} \rangle], \\ & [\text{inequality: } \langle \text{linearConstraints} \rangle] \end{aligned}$$

Both the equality and inequality constraints are expressed through a matrix which is $c \times n_c$ where c is the number of constraints, and a vector which is $c \times 1$. Consequently we have

$$\begin{aligned} \langle \text{linearConstraints} \rangle \rightarrow & \text{matrix: } \langle \text{doubleMatrix} \rangle, \\ & \text{vector: } \langle \text{doubleVector} \rangle \end{aligned}$$

A $\langle \text{doubleMatrix} \rangle$ is defined by

$$\langle \text{doubleMatrix} \rangle \rightarrow \langle \text{doubleVector} \rangle, \dots, \langle \text{doubleVector} \rangle$$

and a $\langle \text{doubleVector} \rangle$ is just what the name suggests – a vector of doubles:

$$\langle \text{doubleVector} \rangle \rightarrow \langle \text{double} \rangle, \dots, \langle \text{double} \rangle.$$

The only remaining undescribed element above is $\langle \text{continuousCapability} \rangle$ whose description is

$$\begin{aligned} \langle \text{continuousCapability} \rangle \rightarrow & \text{breakPoints: } \langle \text{doubleListMatrix} \rangle, \\ & \text{valAtBreakPoints: } \langle \text{doubleListMatrix} \rangle \end{aligned}$$

$\langle \text{doubleListMatrix} \rangle$ describes a $n_c \times n_c$ matrix whose elements are lists of $\langle \text{double} \rangle$:

$$\begin{aligned} \langle \text{doubleListMatrix} \rangle \rightarrow & \langle \text{doubleListVector} \rangle, \dots, \langle \text{doubleListVector} \rangle \\ \langle \text{doubleListVector} \rangle \rightarrow & \langle \text{doubleList} \rangle, \dots, \langle \text{doubleList} \rangle \\ \langle \text{doubleList} \rangle \rightarrow & \{ \langle \text{double} \rangle \} \end{aligned}$$

It is assumed that the rows and columns of the matrix are in some canonical order so that we know which continuous variable is referenced. A natural order is the one defined in $\{ \langle \text{continuousVarDescription} \rangle \}$

Preferences

Just as capabilities may be buyer-specific so too may preferences be seller-specific. The same rules determining which seller-specific preference to apply are followed. A schematic of a $\langle \text{sellerSpecificPreference} \rangle$ is given in Figure 3.

We define a preference as follows

$$\langle \text{preference} \rangle \rightarrow \{ \langle \text{sellerSpecificPreference} \rangle \}, \langle \text{aggregatedPreference} \rangle$$

i.e., a preference is a list of $\langle \text{sellerSpecificPreference} \rangle$ with an optional aggregated preference. We first describe $\langle \text{sellerSpecificPreference} \rangle$ and then consider $\langle \text{aggregatedPreference} \rangle$.

The $\langle \text{sellerSpecificPreference} \rangle$ is composed as follows

$$\begin{aligned} \langle \text{sellerSpecificPreference} \rangle \rightarrow & \text{id: } \langle \text{id} \rangle, \\ & \text{discrete: } \{ \langle \text{discreteVarDescription} \rangle \}, \\ & \text{continuous: } \{ \langle \text{continuousVarDescription} \rangle \}, \\ & \text{range: } \{ \langle \text{rangeVarDescription} \rangle \}, \\ & \text{dimensionWeights: } \langle \text{dimensionWeights} \rangle, \\ & \text{discreteTradeoff: } \langle \text{tradeoffTables} \rangle \\ & [\text{discreteConstraint: } \langle \text{discreteConstraint} \rangle], \\ & \text{instance: } \{ \langle \text{discretePreferenceInstance} \rangle \} \end{aligned}$$

Of these elements, only $\langle \text{dimensionWeights} \rangle$, $\langle \text{tradeoffTables} \rangle$, and $\langle \text{discretePreferenceInstance} \rangle$ have yet to be defined. $\langle \text{dimensionWeights} \rangle$ gives the weights of all dimensions that indicate their importance. For convenience we break up the weights according to the three types of variables. Thus we have

$$\begin{aligned} \langle \text{dimensionWeights} \rangle \rightarrow & \text{range: } \langle \text{doubleVector} \rangle, \\ & \text{discrete: } \langle \text{doubleVector} \rangle, \\ & \text{continuous: } \langle \text{doubleVector} \rangle \end{aligned}$$

A $\langle \text{doubleVector} \rangle$ has been described previously. Each of the corresponding vectors is as long as the number of range, discrete, or continuous dimensions. $\langle \text{tradeoffTables} \rangle$ is an $n_{\text{discrete}} \times n_{\text{discrete}}$ matrix of $\langle \text{tradeoffTable} \rangle$:

$$\begin{aligned} \langle \text{tradeoffTables} \rangle & \rightarrow \langle \text{tradeoffTableMatrix} \rangle \\ \langle \text{tradeoffTableMatrix} \rangle & \rightarrow \langle \text{tradeoffTableVector} \rangle, \dots, \langle \text{tradeoffTableVector} \rangle \\ \langle \text{tradeoffTableVector} \rangle & \rightarrow \langle \text{tradeoffTable} \rangle, \dots, \langle \text{tradeoffTable} \rangle \end{aligned}$$

A $\langle \text{tradeoffTable} \rangle$ is simply a matrix of double values.

Finally, we turn to the last undefined component of a $\langle \text{preference} \rangle$. A $\langle \text{discretePreferenceInstance} \rangle$ is

composed as follows:

```

<discretePreferenceInstance> → mask: <mask>,
                                [rangeIdeal: <{<rangeVarInstance>}>],
                                continuousIdeal: <doubleVector>,
                                tradeoffMatrix: <doubleMatrix>,
                                [continuousConstraints: <continuousConstraints>]

```

The rangeIdeal and continuousIdeal fields give the desired range and continuous trade parameters. The tradeoffMatrix field gives the positive definite matrix expressing the tradeoffs amongst the continuous variables. <continuousConstraints> have been described previously in the sell-side specification.

To complete the specification of preferences, we conclude with the definition of <aggregatedPreference>. Refer to the discussion of section 6.1.7 for details.

```

<aggregatedPreference> → participants: <{<aggSpecification>}>,
                        contributionType: <contributionTypeVector>,
                        additionalConstraints: <continuousConstraints>,
                        discrete: <{<discreteVarDescription>}>,
                        continuous: <{<continuousVarDescription>}>,
                        range: <{<rangeVarDescription>}>,
                        dimensionWeights: <dimensionWeights>,
                        discreteTradeoff: <tradeoffTables>
                        [discreteConstraint: <discreteConstraint>],
                        instance: <{<discretePreferenceInstance>}>

```

In the above definition, the previously defined elements maintain their meaning. The additionalConstraints field allows the buyer to express constraints around the aggregation, such as "all orders must arrive on the same day," etc. participants lists the suppliers who can participate in the aggregation and their characteristics. Note that if the wildcard supplier participates, the order can potentially be aggregated across all suppliers. <aggSpecification> describes information specific to a supplier participating in the aggregation. It is defined by

```

<aggSpecification> → id: <id>.

```

id identifies the participating supplier and constraints specific to that supplier defined in an accompanying <sellerSpecificPreference> will be used in the optimization. Additional information may be added as required.

discrete mask	output	specificity
$\begin{bmatrix} * & * & * \end{bmatrix}$	{continuous1, range1}	0
$\begin{bmatrix} \kappa_1 & * & * \end{bmatrix}$	{continuous2, range2}	1
$\begin{bmatrix} \kappa_1 & * & \kappa_3 \end{bmatrix}$	{continuous3, range3}	2

Table 3: Example of discrete masks for specifying continuous and range variables which are dependent on discrete variables. κ_1 and κ_3 signify specific values for the first and third discrete variables. The specificity of each mask is indicated in the third column.

The contributionType field is used to define the ξ vectors used in aggregation. The $\langle \text{contributionTypeVector} \rangle$ consists of n_c elements indicating the type of aggregation for each continuous dimension:

$$\langle \text{contributionTypeVector} \rangle \rightarrow \langle \text{contributionType} \rangle, \dots, \langle \text{contributionType} \rangle.$$

Possible contribution types include

$$\langle \text{contributionType} \rangle \rightarrow \text{sum, average, zero.}$$

sum sets $\xi = 1$, average sets $\xi = 1/|\mathcal{K}(\kappa)|$, and zero sets $\xi = 0$.

Masking

We have allowed constraints, ideal values, tradeoffs, and continuous capabilities to be dependent on discrete variables. In this section we describe an efficient manner in which to encode this dependence.

The data structure must represent continuous and range variables for all valid discrete inputs. An efficient way to do this is to use hierarchical definitions. At the top of the hierarchy are the definitions of the continuous and range variables for the discrete values $\kappa^t = [*, \dots, *]$. These values apply to all v unless more specialized masks are defined. A more specialized mask of the continuous and range variables is specified by defining values for some of the components κ_i . The more components that are defined, the more specialized the definition. The most specific mask is always used. An example definition for three discrete variables is given in Table 3. The response to the lookup $[\tilde{\kappa}_1 \ \tilde{\kappa}_2 \ \tilde{\kappa}_3]$ is {continuous3, range3} if and only if $\kappa_1 = \tilde{\kappa}_1 \wedge \kappa_3 = \tilde{\kappa}_3$, {continuous2, range2} if and only if $\kappa_1 = \tilde{\kappa}_1 \wedge \kappa_3 \neq \tilde{\kappa}_3$, and {continuous1, range1} otherwise.

Match Results

Returned to the buyer is a list of matches with different suppliers, which can be ranked and viewed in many different ways in the GUI. A $\langle \text{matchResultList} \rangle$ is a list of matchResult:

$$\langle \text{matchResultList} \rangle \rightarrow \{ \langle \text{matchResult} \rangle \}.$$

A match result may either be a `<singleSupplierMatchResult>` or an `<aggregatedMatchResult>`:

`<matchResult> → <singleSupplierMatchResult>|<aggregatedMatchResult>`.

A `<singleSupplierMatchResult>` represents the best trade with a single supplier and is composed of the following elements:

`<singleSupplierMatchResult> → supplierId: <integer>,
utility: <double>,
feasible: 0|1,
[costFactors: {<double>}],
continuous: {<double>},
discrete: {<discreteVarDescription>},
range: <rangeVarInstance>.`

The `supplierId` indicates the supplier sourcing this trade and the `utility` field indicates the utility of the trade (which can be used to rank the trades). `feasible` is a bit indicating whether or not a feasible trade with this supplier was found. The `continuous`, `discrete`, and `range` fields list the respective trade parameters determined by the matching algorithm. The optional `costFactors` field lists the constituent costs contributing to the total cost of ownership C_o evaluated at the trade point returned in the `<singleSupplierMatchResult>`.

An `<aggregatedMatchResult>` returns the optimal trade when the buyer has requested aggregation. It is composed of the following elements:

`<aggregatedMatchResult> → utility: <double>,
feasible: 0|1,
[costFactors: {<double>}],
supplierTradeParameters: {<supplierTradeParameters>}.`

As before, the `utility` field gives the utility of the aggregate trade, and the `feasibility` flag indicates whether or not a feasible aggregate trade was found (there may not be if the constraints were too stringent). `costFactors` can also be returned in C_o is sufficiently general to handle aggregated trades. Finally, `<supplierTradeParameters>` lists the trade parameters for each supplier involved in the aggregation. It is defined as follows:

`<supplierTradeParameters> → supplierId <integer>,
continuous: {<double>},
discrete: {<discreteVarDescription>},
range: <rangeVarInstance>.`

6.3 Summary

We have described an efficient computational procedure in which to encode buyer's trading preferences and hard constraints, supplier's delivery capabilities and constraints, and optimize to find those matches between one buyer and one or many sellers that maximize the buyer's utility. By optimizing against both qualitative and quantitative factors, and exploiting the trading flexibilities possessed by both parties, the system determines better trades. The tool is particularly useful as companies move their direct material purchasing online. By optimizing across flexibilities, win-win trades are discovered for both trading parties.

The representation of trading preferences is designed to be expressive yet easily elicitable from a buyer, and computationally tractable. The representation of supplier capabilities was chosen to parallel the manner in which suppliers already think of their delivery capabilities and seamlessly includes volume discounts and incremental costs. These supplier capabilities may be part of an online catalog. The representation of the negotiation space is rich, supporting three types of variables.

We have outlined a manner in which preferences may be inferred automatically through models of the purchasing company. Such models incorporate many cost factors, taking the total cost of ownership into account. The system provides trades which minimize the total cost and represent significant new savings.

The invention can operate both at a buyer's site, where suppliers input their capabilities through an HTML interface to the world wide web or as an embedded part of an electronic market hosted by a particular web site. The invention may operate at regularly scheduled intervals or sporadically in lieu of current request for quotations (RFQ). The buyer may broadcast a RFQ event to suppliers, indicating a time within which suppliers must respond. At the close of the event, the buyer can use the present invention to assist in the analysis of the supplier responses.

Complex algorithms have been specified which should permit most matching optimization to occur in near real-time. The rapidity of optimization, combined with graphical what-if tools, allows for analysis and exploration of trades, which should significantly improve the quality of purchasing decisions.

While the above invention has been described with reference to certain preferred embodiments, the scope of the present invention is not limited to these embodiments. One skilled in the art may find variations of these preferred embodiments which, nevertheless, fall within the spirit of the present invention, whose scope is defined by the claims set forth below.

7 Claims

What is claimed in the present invention is

1. A method for discovery of trades between one or more buyers and one or more sellers comprising the steps of:
 - (a) expressing one or more terms of an ideal trade and one or more flexibilities by at least one of said buyers;
 - (b) expressing one or more capabilities by at least one of said sellers; and
 - (c) determining at least one optimal trade with respect to said one or more terms and said one or more flexibilities of said at least one buyer and said one or more capabilities of said at least one seller.
2. A method for discovery of trades between one or more buyers and one or more sellers as in 1, wherein said one or more terms comprise one or more members of the group consisting of continuous factors, discrete factors and range factors.

8 References

- [1] Keeney, Ralph L., Raiffa, Howard, *Decisions with Multiple Objectives*, Cambridge University Press:Cambridge, 1993.
- [2] Saaty, Thomas L., *The Analytic Network Process*, RWS Publications:Pittsburg, 1996.
- [3] A concise introduction to discrete choice analysis with a comparison to conjoint analysis is found at <http://www.action-research.com/discrete.htm>.
- [4] Nesterov, Yurii, Nemirovskii, Arkadii, *Interior Point Polynomial Algorithms in Convex Programming*, Society for Industrial and Applied Math:Philadelphia, 1994.
- [5] Tsang, Edward, *Foundations of Constraint Satisfaction*, Academic Press:London, 1993.

A Distance Function for Range Variables

The distance between buyer range variable $r_i = (\underline{r}_i, \bar{r}_i)$ and seller range variable $r_j = (\underline{r}_j, \bar{r}_j)$ is related to the overlap defined as

$$\text{overlap}(r_i, r_j) = \int_{\mu_j - \sigma_j}^{\mu_j + \sigma_j} dx \mathcal{N}(x; r_i) \mathcal{N}(x; r_j)$$

where $\mathcal{N}(x; r_i)$ is a Gaussian in x centered at $\mu_i = (\underline{r}_i + \bar{r}_i)/2$ with standard deviation $\sigma_i = \alpha(\bar{r}_i - \underline{r}_i)$ and α is a tunable parameter. The range of integration is bounded by the seller's capabilities. Evaluating the integral, we find

$$\text{overlap}(r_i, r_j) = \frac{1}{2\sqrt{2\pi(\sigma_i^2 + \sigma_j^2)}} \exp \left[-\frac{(\mu_i - \mu_j)^2}{2(\sigma_i^2 + \sigma_j^2)} \right] \left(\text{erf} \left[\frac{\sigma_j^2 + \sigma_j\mu_i - \sigma_j\mu_j + \sigma_i^2}{\sigma_i\sqrt{2(\sigma_i^2 + \sigma_j^2)}} \right] - \text{erf} \left[\frac{-\sigma_j^2 + \sigma_j\mu_i - \sigma_j\mu_j - \sigma_i^2}{\sigma_i\sqrt{2(\sigma_i^2 + \sigma_j^2)}} \right] \right).$$

By adjusting α we can control the hardness of the boundary at the ends of the range (smaller α corresponds to sharper dropoff outside the range). The maximum overlap will occur when $\mu_j = \mu_i$ and $\sigma_j = 0$. Substituting those values in the above equation, we get

$$\text{maxOverlap} = \frac{\text{erf} \left[\frac{1}{\sqrt{2}} \right]}{\sqrt{2\pi(\sigma_i^2 + \sigma_j^2)}}$$

The distance $d(r_i, r_j)$ is related to the overlap so that it varies from 0 (for identical range variables) to ∞ (for maximally distant range variables). Consequently, we define

$$R(r_i, r_j) = -\ln \left[\frac{\text{overlap}(r_i, r_j)}{\text{maxOverlap}} \right].$$

B From Rankings to Numerical Values

The numerical values in the tables $Z(x_i, x_j)$ can be formed from the buyer's rankings in a number of ways. Since we are minimizing distance, we want preferred options to contribute little Z and progressively less favorable options to have progressively larger Z values. To approach this, we first consider the problem of defining a Z' value normalized to lie between 0 and 1 and then take $Z = -\ln[1 - Z']$.

One possibility for Z' is a linear scaling. If d_i is the integer indicating the number of values in \mathcal{D}_i , $r_i^{(\alpha)}(x_j)$ indicates the buyer-specified rank (smaller rank means more preferable) for the option $x_i = \alpha \in \mathcal{D}_i$ conditional on x_j , then we might set $Z'(x_i = \alpha, x_j) = (r_i^{(\alpha)}(v_j) - 1)/(d_i - 1)$. Alternatively, we might use a non-linear scaling and set $Z(x_i = \alpha, v_j) = (1 - \beta^{r_i^{(\alpha)}(x_j) - 1})/(1 - \beta^{d_i - 1})$ for some $\beta \in [0, 1]$ (exponential) or set $Z(v_i = \alpha, v_j) = ((r_i^{(\alpha)}(v_j) - 1)/(d_i - 1))^n$ for some $n > 0$ (algebraic). The corresponding scaling functions for Z are given in Table 4.

type	Z'
linear	$\ln \left[\frac{R_i - 1}{R_i - r_i^{(\alpha)}(v_j)} \right]$
exponential	$\ln \left[\frac{1 - \beta^{R_i-1}}{\beta^{r_i^{(\alpha)}(v_j)-1} - \beta^{R_i-1}} \right]$
algebraic	$-\ln \left[1 - \left(\frac{r_i^{(\alpha)}(v_j) - 1}{R_i - 1} \right)^n \right]$

Table 4: Possible numerical ranking functions.

For those allowed values of κ_i which are left unranked by the buyer, we can assign them a common numerical value greater than the lowest ranked value. For example if $r_i(\kappa_j)$ is the largest ranked value we might assign all remaining κ_i values to have $Z(\kappa_i, \kappa_j) = r_i(\kappa_j)/(d_i - 1)$ in the linear case or $Z(\kappa_i, \kappa_j) = (1 - \beta^{r_i(\kappa_j)})/(1 - \beta^{d_i-1})$ or $Z(\kappa_i, \kappa_j) = (1 - 1/(r_i(\kappa_j) + 1))/(1 - 1/d_i)$ in the non-linear cases.

C Determination of Scaling Factors

In this section we supply the details necessary to determine the scaling factors Q_r and Q_d . In the equations defining the average distances we have need of the integrals $\int_V du \exp[-(u - \bar{u})^t C^{-1} (u - \bar{u})]$ and $\int_V du (u - \bar{u})^t C^{-1} (u - \bar{u}) \exp[-(u - \bar{u})^t C^{-1} (u - \bar{u})]$. In general, these integrations are not analytically tractable; however we will assume that the region of integration defined by \bar{x} and \underline{x} extends at least 2 standard deviations out in all directions so that the volume of integration is effectively infinite. In this case we can evaluate the integrals in closed form:

$$\int_V du \exp[-(u - \bar{u})^t C^{-1} (u - \bar{u})] \approx \sqrt{\frac{\pi^{n_c}}{\det C^{-1}}}$$

and

$$\int_V du (u - \bar{u})^t C^{-1} (u - \bar{u}) \exp[-(u - \bar{u})^t C^{-1} (u - \bar{u})] \approx \frac{n_c}{2} \sqrt{\frac{\pi^{n_c}}{\det C^{-1}}}.$$

To this point we have not been specific about what is meant by the average over the range variables. In fact, we will assume that, however we define this average, it is independent of what the buyer requested and thus can be removed from inside the sum over the discrete variables. In this case we will concentrate on setting the discrete and continuous components equal. We have $\langle d_c \rangle / n_c = \langle d_d \rangle / n_d$ where

$$\frac{\langle d_d \rangle}{n_d} \equiv \frac{Q_d \sum_v d_d \exp[-Q_d d_d] \int_V du \exp[-d_c]}{\sum_v \exp[-Q_d d_d] \int_V du \exp[-d_c]} \quad \text{and} \quad \frac{\langle d_c \rangle}{n_c} \equiv \frac{1 \sum_v \exp[-Q_d d_d] \int_V du d_c \exp[-d_c]}{\sum_v \exp[-Q_d d_d] \int_V du \exp[-d_c]}$$

and cancelling denominators Q_d is determined by solving

$$\frac{Q_d}{n_d} \sum_v d_d \exp[-Q_d d_d] \int_V du \exp[-d_c] = \frac{1}{n_c} \sum_v \exp[-Q_d d_d] \int_V du d_c \exp[-d_c].$$

This can be simplified further given our assumptions about the Gaussian integral:

$$\frac{Q_d}{n_d} \sum_{\mathbf{v}} \frac{d_d \exp[-Q_d d_d]}{\sqrt{\det \mathbf{C}^{-1}(\mathbf{v})}} = \frac{1}{2} \sum_{\mathbf{v}} \frac{\exp[-Q_d d_d]}{\sqrt{\det \mathbf{C}^{-1}(\mathbf{v})}}.$$

In most cases there will also be at least one finite root, which is what we need. If there are a multiplicity of roots, we select that root having the smallest value to ensure that the distribution defined by the utility function is as flat as possible. If there are no real roots, then we select the Q_d which minimizes the error.

D Phase 1 Optimization

In this section we detail the solution to the phase 1 optimization problem defined in Eq. (5).

D.1 General Formulation

We begin by writing down the objective function:

$$d_1(\mathbf{x}, \boldsymbol{\kappa}) = (\mathbf{x}^{(s)} - \boldsymbol{\mu}(\boldsymbol{\kappa}))^t \mathbf{C}_w^{-1} (\mathbf{x}^{(s)} - \boldsymbol{\mu}(\boldsymbol{\kappa})) + R(\mathbf{r}; \boldsymbol{\kappa})$$

where (see Eq. (8))

$$\mathbf{x}^{(s)} = (\mathbf{I} - \mathbf{M}^{(s)}(\boldsymbol{\kappa}^{(s)}, \{b_{i,k}^{(s)}\}))^{-1} \mathbf{c}(\boldsymbol{\kappa}^{(s)}, \{b_{i,k}^{(s)}\}, \{b_{i,k}^{(b)}\}) + (\mathbf{I} - \mathbf{M}^{(s)}(\mathbf{v}^{(s)}, \{b_{i,k}^{(s)}\}))^{-1} \mathbf{M}^{(b)}(\{b_{i,k}^{(b)}\}) \mathbf{u}^{(b)} \quad (8)$$

as long as $x_k^{(s)}(b_{i,k}^{(s)}) \leq x_k^{(s)} \leq x_k^{(s)}(b_{i,k}^{(s)} + 1)$. This must be optimized over \mathbf{x} subject to the constraints $\mathbf{G}_1(\boldsymbol{\kappa})\mathbf{x}^{(s)} = \mathbf{g}_1(\boldsymbol{\kappa})$ and $\mathbf{G}_2(\boldsymbol{\kappa})\mathbf{x}^{(s)} = \mathbf{g}_2(\boldsymbol{\kappa})$ for a given $\boldsymbol{\kappa}$. Since $\boldsymbol{\kappa}$ is fixed throughout the optimization $R(\mathbf{r}; \boldsymbol{\kappa})$ is just a constant which can be evaluated and then ignored as far as determination of \mathbf{x} is concerned. We simplify our notation so that $d_1(\mathbf{u}) = (\mathbf{x}^{(s)} - \boldsymbol{\mu})^t \mathbf{C}_w^{-1} (\mathbf{x}^{(s)} - \boldsymbol{\mu}) + R(\mathbf{r})$. Similarly,

$$\begin{aligned} \mathbf{x}^{(s)} &= (\mathbf{I} - \mathbf{M}^{(s)}(\{b_{i,k}^{(s)}\}))^{-1} \mathbf{c}(\{b_{i,k}^{(s)}\}, \{b_{i,k}^{(b)}\}) + (\mathbf{I} - \mathbf{M}^{(s)}(\{b_{i,k}^{(s)}\}))^{-1} \mathbf{M}^{(b)}(\{b_{i,k}^{(b)}\}) \mathbf{x}^{(b)} \\ &\equiv \mathbf{s}(\{b_{i,k}^{(s)}\}, \{b_{i,k}^{(b)}\}) + \mathbf{S}(\{b_{i,k}^{(s)}\}, \{b_{i,k}^{(b)}\}) \mathbf{x}^{(b)} \end{aligned}$$

where we have defined the n_c -vector $\mathbf{s} = (\mathbf{I} - \mathbf{M}^{(s)})^{-1} \mathbf{c}$ and the $n_c \times n'_c$ matrix $\mathbf{S} = (\mathbf{I} - \mathbf{M}^{(s)})^{-1} \mathbf{M}^{(b)}$.¹⁶ $n'_c < n_c$ is the number of continuous dimensions upon which the supplier response depends. Putting these pieces together we have

$$d_1(\mathbf{x}^{(b)}) = \mathbf{x}^{(b)t} \mathbf{Q}(\{b_{i,k}^{(s)}\}, \{b_{i,k}^{(b)}\}) \mathbf{x}^{(b)} + \mathbf{q}^t \mathbf{x}^{(b)} + \text{const} \quad (9)$$

where \mathbf{Q} is an $n'_c \times n'_c$ matrix given by

$$\mathbf{Q}(\{b_{i,k}^{(s)}\}, \{b_{i,k}^{(b)}\}) = \mathbf{S}^t(\{b_{i,k}^{(s)}\}, \{b_{i,k}^{(b)}\}) \mathbf{C}_w^{-1} \mathbf{S}(\{b_{i,k}^{(s)}\}, \{b_{i,k}^{(b)}\})$$

¹⁶In the next section we consider \mathbf{S} further, taking advantage of any redundant variables to simplify the QP task.

q is an n' -vector given by

$$q(\{b_{i,k}^{(s)}\}, \{b_{i,k}^{(b)}\}) = 2S^t(\{b_{i,k}^{(s)}\}, \{b_{i,k}^{(b)}\})C_w^{-1}(s(\{b_{i,k}^{(s)}\}, \{b_{i,k}^{(b)}\}) - \mu)^t.$$

and the constant term is

$$const = (s(\{b_{i,k}^{(s)}\}, \{b_{i,k}^{(b)}\}) - \mu)^t C_w^{-1}(s(\{b_{i,k}^{(s)}\}, \{b_{i,k}^{(b)}\}) - \mu) + R(r)$$

It is easily shown that the matrix Q is positive semidefinite. $d_1(u^{(b)})$ must be optimized subject to the constraints

$$G_1 S(\{b_{i,k}^{(s)}\}, \{b_{i,k}^{(b)}\})x^{(b)} = g_1 - G_1 s(\{b_{i,k}^{(s)}\}, \{b_{i,k}^{(b)}\}), \quad (10)$$

$$G_2 S(\{b_{i,k}^{(s)}\}, \{b_{i,k}^{(b)}\})x^{(b)} \leq g_2 - G_2 s(\{b_{i,k}^{(s)}\}, \{b_{i,k}^{(b)}\}), \quad (11)$$

$$\underline{x}^{(s)}(i) \leq s(\{b_{i,k}^{(s)}\}, \{b_{i,k}^{(b)}\}) + S(\{b_{i,k}^{(s)}\}, \{b_{i,k}^{(b)}\})x^{(b)} \leq \bar{x}^{(s)}(i) \quad \forall i, \quad (12)$$

$$\underline{x}^{(b)}(i) \leq x^{(b)} \leq \bar{x}^{(b)}(i) \quad \forall i \quad (13)$$

where $\underline{x}^{(s)}(i)^t = [x_1(b_{i,1}^{(s)}) \dots x_n(b_{i,n}^{(s)})]$ and $\bar{x}^{(s)}(i)^t = [x_1(b_{i,1}^{(s)} + 1) \dots x_n(b_{i,n}^{(s)} + 1)]$ are the vectors of breakpoints for the $f_{i,k}^{(s)}$ ($k \in [1, n_c]$) functions. Similarly, for the buyer variables, $\underline{x}^{(b)}(i)^t = [x_1(b_{i,1}^{(b)}) \dots x_n(b_{i,n}^{(b)})]$ and $\bar{x}^{(b)}(i)^t = [x_1(b_{i,1}^{(b)} + 1) \dots x_n(b_{i,n}^{(b)} + 1)]$ are the vectors of breakpoints for the $f_{i,k}^{(b)}$ ($k \in [1, n_c]$) functions.

d_1 is optimized in two steps. First we assume a value for the integer variables $\{b_{i,k}^{(s)}\}$ and $\{b_{i,k}^{(b)}\}$ fixing the constraints in Eqs. (10) and (11). We then minimize the quadratic d_1 subject to these constraints using a standard quadratic programming (QP) algorithm¹⁷ to find $x_{opt}^{(b)}(\{b_{i,k}^{(s)}\}, \{b_{i,k}^{(b)}\})$. This result is then used to find $x^*(x; \{b_{i,k}^{(s)}\}, \{b_{i,k}^{(b)}\}) = s(\{b_{i,k}^{(s)}\}, \{b_{i,k}^{(b)}\}) + S(\{b_{i,k}^{(s)}\}, \{b_{i,k}^{(b)}\})x_{opt}^{(b)}(\{b_{i,k}^{(s)}\}, \{b_{i,k}^{(b)}\})$. By searching exhaustively over $\{b_{i,k}^{(s)}\}$ and $\{b_{i,k}^{(b)}\}$ we determine

$$x^{(s)}(x) = \arg \min_{\{b_{i,k}^{(s)}\}, \{b_{i,k}^{(b)}\}} d_1(x^*(x; \{b_{i,k}^{(s)}\}, \{b_{i,k}^{(b)}\}), x)$$

Before turning to the manner in which this optimization is accomplished in section D.3, we discuss a few improvements to the input into the QP algorithm.

D.2 The Quadratic Program

In this section we discuss improvements in the specification of the quadratic programming task to make it better conditioned and more efficient.

It is possible that the matrix Q used as input to the QP is singular or near singular, depending on the rank of S_1 . Consequently, it is important to use an interior point QP algorithm for which this is not a problem.

¹⁷Numerous third parties supply quadratic programming packages, some based on interior-point methods which are appropriate, in the present context where matrix Q may be singular.

D.2.1 Variable Elimination

There may be rows or columns of S which are entirely zero. It is important to take advantage of this structure to eliminate unnecessary computation.

If row i is zero, this indicates that supplier variable $u_i^{(s)}$ is independent of any buyer variable and is simply equal to s_i . If there are any constraints involving only supplier variable i , then these can be checked before calling the QP solver to eliminate wasteful computation.¹⁸ If the constraint is satisfied, we can eliminate it and then call the QP function. If column j of S is all zero, this means that there is no dependence of $u^{(s)}$ on $u_j^{(b)}$ and so there is no need to optimize over that buyer variable. We will, however, still have to check that the constraints on $u^{(s)}$ are satisfied. This observation allows us to simplify the quadratic programming problem by eliminating this unnecessary variable.

We formulate the problem to take advantage of both these observations. Assume there are $m < n_c$ non-zero rows of S and $m' < n'_c$ non-zero columns. Relabel the rows and columns so that

$$S = \begin{bmatrix} S_1 & 0 \\ 0 & 0 \end{bmatrix}$$

where S_1 is of size $m \times m'$ and, with a slight abuse of notation, the 0's indicate matrices of the appropriate size. In like fashion we write

$$x^{(s)} = \begin{bmatrix} x_1^{(s)} \\ x_2^{(s)} \end{bmatrix}, \quad x^{(b)} = \begin{bmatrix} x_1^{(b)} \\ x_2^{(b)} \end{bmatrix}$$

where $x_1^{(s)}$ is an m vector, $x_2^{(s)}$ is an $n_c - m$ vector, $x_1^{(b)}$ is an m' vector, and $x_2^{(b)}$ is an $n'_c - m'$ vector. $x_1^{(s)}$ indicates those components of $x^{(s)}$ which vary with buyer variables while $x_2^{(s)}$ indicates those variables which are constant. $x_1^{(b)}$ are those buyer variables which affect at least one supplier variable and $x_2^{(b)}$ are those buyer variables which do not affect any supplier variables. If we also partition s as $s^t = \begin{bmatrix} s_1^t & s_2^t \end{bmatrix}$ then

$$x_1^{(s)} = s_1 + S_1 x_1^{(b)} \quad \text{and} \quad x_2^{(s)} = s_2.$$

Finally, we will also have need of the partitioned covariance matrix

$$C^{-1} = \begin{bmatrix} C_{1,1}^{-1} & C_{1,2}^{-1} \\ C_{2,1}^{-1} & C_{2,2}^{-1} \end{bmatrix}$$

where $(C_{1,2;w}^{-1})^t = C_{2,1;w}^{-1}$ since C_w^{-1} is symmetric.

Including the dependencies on μ , the quadratic form we must optimize is

$$\begin{bmatrix} (s_1 - \mu_1)^t + (x_1^{(b)})^t S_1^t & (s_2 - \mu_2)^t \end{bmatrix} \begin{bmatrix} W_1 C_{1,1}^{-1} W_1 & W_1 C_{1,2}^{-1} W_2 \\ W_2 C_{2,1}^{-1} W_1 & W_2 C_{2,2}^{-1} W_2 \end{bmatrix} \begin{bmatrix} s_1 - \mu_1 + S_1 u_1^{(b)} \\ s_2 - \mu_2 \end{bmatrix}$$

¹⁸Additionally, the QP doesn't like to have a row of zeros in the constraint matrix.

where we have made the obvious definitions

$$W = \begin{bmatrix} W_1 & 0 \\ 0 & W_2 \end{bmatrix} \quad \text{and} \quad \mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}.$$

Expanding the quadratic, the constant term is

$$(s_1 - \mu_1)^t W_1 C_{1,1}^{-1} W_1 (s_1 - \mu_1) + (s_2 - \mu_2)^t W_2 C_{2,2}^{-1} W_2 (s_2 - \mu_2) + 2(s_1 - \mu_1)^t W_1 C_{1,2}^{-1} W_2 (s_2 - \mu_2).$$

The linear term is

$$q = ((s_1 - \mu_1)^t W_1 C_{1,1}^{-1} W_1 S_1^t + 2(s_2 - \mu_2)^t W_2 C_{2,1}^{-1} W_1 S_1) x_1^{(b)}$$

and the quadratic term is

$$Q = (x_1^{(b)})^t S_1^t W_1 C_{1,1}^{-1} W_1 S_1 x_1^{(b)}.$$

The constraints involving the optimization variables $u_1^{(b)}$ are

$$G_{1,1} S_1 x_1^{(b)} = g_1 - G_{1,1} s_1 - G_{1,2} s_2,$$

$$G_{2,1} S_1 x_1^{(b)} \leq g_2 - G_{2,1} s_1 - G_{2,2} s_2,$$

$$\underline{x}_1^{(s)}(i) \leq s_1 + S_1 x_1^{(b)} \leq \bar{x}_1^{(s)}(i) \quad \forall i,$$

$$\underline{x}_1^{(b)}(i) \leq x_1^{(b)} \leq \bar{x}_1^{(b)}(i) \quad \forall i$$

where we have simplified notation by dropping the dependence on breakpoints and defined

$$G_i = \begin{bmatrix} G_{i,1} & G_{i,2} \end{bmatrix}$$

i.e., $G_{i,1}$ is the columns of G_1 involving the active buyer variables while $G_{i,2}$ is formed from the inactive buyer variables. Similarly, the constraints which can be checked prior to even solving the QP are

$$\underline{x}_2^{(s)}(i) \leq s_2 \leq \bar{x}_2^{(s)}(i) \quad \forall i.$$

D.2.2 QP Normalization

There are two types of normalizations we perform for increased numerical stability. If the scales of the continuous dimensions vary greatly (i.e., one dimension has an ideal value of 10^{-2} and another dimension has ideal value 10^6), then the matrix C^{-1} becomes badly conditioned, which makes S badly conditioned and thus makes the QP unstable. Similarly, if the breakpoints have radically different lengths, the QP problem can also become unstable. We consider how to remedy each of these problems.

Dimension normalization

The quadratic form to be minimized is $(\mathbf{x} - \boldsymbol{\mu})^t \mathbf{C}^{-1} (\mathbf{x} - \boldsymbol{\mu})$. We modify this problem by defining the dimensionless variables ϕ_i through $x_i = \phi_i \mu_i$ in terms of which

$$(\mathbf{x} - \boldsymbol{\mu})^t \mathbf{C}^{-1} (\mathbf{x} - \boldsymbol{\mu}) = (\boldsymbol{\phi} - \mathbf{1})^t \mathbf{C}^{-1} (\boldsymbol{\phi} - \mathbf{1})$$

where $\mathbf{C}^{-1} = \mathbf{D} \mathbf{C}^{-1} \mathbf{D}$ and $\mathbf{D} = \text{diag}(\boldsymbol{\mu})$. When making this variable transformation we must assure that the constraints on \mathbf{x} are adjusted so that $\mathbf{G}_1 \mathbf{x} = \mathbf{g}_1$ and $\mathbf{G}_2 \mathbf{x} \leq \mathbf{g}_2$ become the constraints $\mathbf{G}_1 \boldsymbol{\phi} = \mathbf{g}_1$ and $\mathbf{G}_2 \boldsymbol{\phi} \leq \mathbf{g}_2$ on $\boldsymbol{\phi}$ where $\mathbf{G}_{1,2} = \mathbf{G}_{1,2} \mathbf{D}$. If there are global bounds of the form $\underline{\mathbf{x}} \leq \mathbf{x} \leq \bar{\mathbf{x}}$, then these are translated to $\underline{\boldsymbol{\phi}} \leq \boldsymbol{\phi} \leq \bar{\boldsymbol{\phi}}$ where $\underline{\boldsymbol{\phi}} = \mathbf{D}^{-1} \underline{\mathbf{x}}$ and $\bar{\boldsymbol{\phi}} = \mathbf{D}^{-1} \bar{\mathbf{x}}$.

In addition to these changes, we also need to modify the capabilities appropriately. Recall Eq. (2) which reads (after dropping the dependence on the discrete variables)

$$x_i^{(s)} = \sum_k f_{i,k}^{(s)}(x_k^{(s)}) + f_{i,k}^{(b)}(x_k^{(b)}). \quad (14)$$

In terms of the dimensionless variables we have

$$\phi_i^{(s)} = \sum_k \frac{f_{i,k}^{(s)}(\phi_k^{(s)} \mu_k)}{\mu_i} + \frac{f_{i,k}^{(b)}(\phi_k^{(b)} \mu_k)}{\mu_i}. \quad (15)$$

Thus, when we use the dimensionless variables, the function $f_{i,k}$ needs to be modified so that the y dimension is divided by μ_i and the x dimension is multiplied by μ_k .

Once the optimization has determined an optimal $\boldsymbol{\phi}$, we return to the original variables through $\mathbf{x} = \mathbf{D} \boldsymbol{\phi}$.

Breakpoint range normalization

For larger problems it is sometimes important to normalize the input variables for the QP solver. This is carried out in order to improve the QP search performance in situations where there is large variation in the range of the segments $\phi_k^{(b)}(b_k) \leq \phi_k^{(b)} \leq \phi_k^{(b)}(b_k + 1)$ and the magnitude of their values. This can be accomplished by introducing the normalization variables $\hat{\phi}_k^{(b)} = (\phi_k^{(b)} - \phi_k^{(b)}(b_k)) / (\phi_k^{(b)}(b_k + 1) - \phi_k^{(b)}(b_k))$. Here $\hat{\phi}_k^{(b)}$ will have minimum value of zero and maximum of one. We can now express $\phi^{(b)}$ as

$$\boldsymbol{\phi}^{(b)} = \boldsymbol{\Lambda} \hat{\boldsymbol{\phi}}^{(b)} + \mathbf{p} \quad (16)$$

where $\boldsymbol{\Lambda}$ is the diagonal matrix $\boldsymbol{\Lambda} = \text{diag}(\phi_1^{(b_1+1)} - \phi_1^{(b_1)}, \phi_2^{(b_2+1)} - \phi_2^{(b_2)}, \dots, \phi_m^{(b_m+1)} - \phi_m^{(b_m)})$ and $\mathbf{p}^t = [\phi_1^{(b_1)}, \phi_2^{(b_2)}, \dots, \phi_m^{(b_m)}]$.

Expanding Eq.(9) using the definition of $\boldsymbol{\phi}$ in Eq.(16), we obtain

$$d_1(\mathbf{w}; \{b_{i,k}\}) = \frac{1}{2} \hat{\boldsymbol{\phi}}^{(b)t} \boldsymbol{\Lambda} \mathbf{D} \mathbf{Q}(\{b_{i,k}\}) \mathbf{D} \boldsymbol{\Lambda} \hat{\boldsymbol{\phi}}^{(b)} + \mathbf{q}^t(\{b_{i,k}\}) \mathbf{D} \boldsymbol{\Lambda} \hat{\boldsymbol{\phi}}^{(b)} + \mathbf{p}^t \mathbf{D} \mathbf{Q}(\{b_{i,k}\}) \mathbf{D} \boldsymbol{\Lambda} \hat{\boldsymbol{\phi}}^{(b)} + \frac{1}{2} \mathbf{p}^t \mathbf{D} \mathbf{Q}(\{b_{i,k}\}) \mathbf{D} \mathbf{p} + \mathbf{q}^t(\{b_{i,k}\}) \mathbf{D} \mathbf{p}.$$

Hence the quadratic programming problem to be solved in terms of $\hat{\boldsymbol{\phi}}$ is

$$QP(\hat{\mathbf{x}}^{(b)}; \{b_{i,k}\}) = \frac{1}{2} \hat{\mathbf{x}}^{(b)t} \hat{\mathbf{Q}}(\{b_{i,k}\}) \hat{\mathbf{x}}^{(b)} + \hat{\mathbf{q}}^t(\{b_{i,k}\}) \hat{\mathbf{u}}^{(b)}, \quad (17)$$

where $\hat{\mathbf{Q}} = \Lambda \mathbf{DQ}(\{b_{i,k}\}) \mathbf{DA}$ and $\hat{\mathbf{q}}^t = (\mathbf{q}^t(\{b_{i,k}\}) + \mathbf{p}^t \mathbf{DQ}(\{b_{i,k}\})) \mathbf{DA}$. Similarly, the equality and inequality constraints of Eqs. (10) and (11) expressed in terms of $\hat{\phi}$ are

$$\hat{\mathbf{G}}_1(\{b_{i,k}\}) \hat{\phi}^{(b)} = \hat{\mathbf{g}}_1 \quad \text{and} \quad \hat{\mathbf{G}}_2(\{b_{i,k}\}) \hat{\phi}^{(b)} \leq \hat{\mathbf{g}}_2, \quad (18)$$

where

$$\begin{aligned} \hat{\mathbf{G}}_1(\{b_{i,k}\}) &= \mathbf{G}_1 \mathbf{S}(\{b_{i,k}\}) \mathbf{DA}, \\ \hat{\mathbf{G}}_2(\{b_{i,k}\}) &= \mathbf{G}_2 \mathbf{S}(\{b_{i,k}\}) \mathbf{DA}, \\ \hat{\mathbf{g}}_1 &= \mathbf{g}_1 - \mathbf{G}_1 \mathbf{s}(\{b_{i,k}\}) - \mathbf{G}_1 \mathbf{S}(\{b_{i,k}\}) \mathbf{Dp}, \\ \hat{\mathbf{g}}_2 &= \mathbf{g}_2 - \mathbf{G}_2 \mathbf{s}(\{b_{i,k}\}) - \mathbf{G}_1 \mathbf{S}(\{b_{i,k}\}) \mathbf{Dp}. \end{aligned}$$

D.3 Optimization Over The Breakpoint Variables

We have detailed how to determine $\mathbf{x}_{\text{opt}}^{(b)}(\mathbf{x}, \{b_{i,k}^{(s)}\}, \{b_{i,k}^{(b)}\})$ given a specification of all $\{b_{i,k}^{(s)}\}$ and $\{b_{i,k}^{(b)}\}$, but how do we determine the best feasible choice of $\{b_{i,k}^{(s)}\}$ and $\{b_{i,k}^{(b)}\}$? Before we answer that question, we have to examine the infeasibilities that might arise due to improper choices of $\{b_{i,k}^{(s)}\}$ and $\{b_{i,k}^{(b)}\}$.

D.3.1 Elimination of $\{b_{i,k}\}$ Infeasibilities

Most of the settings of $\{b_{i,k}^{(b)}\}$ or $\{b_{i,k}^{(s)}\}$ will not yield a feasible solution because Eq. (13) or Eq. (12) will be violated. The reason is that, for any k , it is highly likely to get two segments $x_k^{(b)}(b_{i,k}) \leq x_k^{(b)} \leq x_k^{(b)}(b_{i,k} + 1)$ and $x_k^{(b)}(b_{j,k}) \leq x_k^{(b)} \leq x_k^{(b)}(b_{j,k} + 1)$ that do not overlap (i.e., the optimizer will not be able to find a $x_k^{(b)}$ value that satisfies both bounds, and hence no feasible solution will be found). The same is also true for supplier response segment $x_k^{(s)}$.

To eliminate those infeasibilities, we construct a super set of breakpoints $\{\omega_k^b\}$ such that

$$\omega_k^{(b)} = \bigcup_{i=1}^n \{x_k^{(b)}(b_{i,k}^{(b)} = 1), x_k^{(b)}(b_{i,k}^{(s)} = 2), \dots, x_k^{(b)}(b_{i,k}^{(b)} = \kappa_{i,k}^{(b)})\}$$

Hence each piecewise linear function $f_{i,k}$ will be described by a set of breakpoints $\omega_k = \{x_k(1), \dots, x_k(\kappa_k)\}$ and function values $f_{i,k}(x_k(1)), \dots, f_{i,k}(x_k(\kappa_k))$. These function values at the new breakpoints are linearly interpolated from the old breakpoint-function pairs. After this reduction we have $\kappa_k - 1$ different $b_k \in [1, \kappa_k - 1]$ values, which eliminates the need for constraint Eq. (13).

The continuous supplier variables and constraint Eq. (12) are more problematic. Suppressing any dependence on \mathbf{x} , we know

$$x_i^{(s)} = c_i(\{b_{i,1}^{(s)}, \dots, b_{i,i-1}^{(s)}\}, \{b_{i,k}^{(b)}\}) + \sum_{k < i} m_{i,k}^{(s)}(b_{i,k}^{(s)}) x_k^{(s)} + \sum_k m_{i,k}^{(b)}(b_{i,k}^{(b)}) x_k^{(b)}.$$

If $\{b_{i,k}^{(b)}\}$ and $\{b_{i,1}^{(s)}, \dots, b_{i,i-1}^{(s)}\}$ are known, then this expression allows us to bound the values $x_i^{(s)}$ may take on, and thus bound the interval in which the feasible $b_{i,i}^{(s)}$ lie. If we define the functions

$$\text{smax}(s, x, y) = \begin{cases} s \max(x, y) & \text{if } s \geq 0 \\ s \min(x, y) & \text{if } s < 0 \end{cases} \quad \text{and} \quad \text{smin}(s, x, y) = \begin{cases} s \min(x, y) & \text{if } s \geq 0 \\ s \max(x, y) & \text{if } s < 0 \end{cases}$$

then the minimum and maximum values that $x_i^{(s)}$ may take on are:

$$\begin{aligned} \min(x_i^{(s)}) &= c_i(\{b_{i,1}^{(s)}, \dots, b_{i,i-1}^{(s)}\}, \{b_{i,k}^{(b)}\}) + \sum_{k < i} \text{smin}(m_{i,k}^{(s)}(b_{i,k}^{(s)}), x_k^{(s)}(b_{i,k}^{(s)}), x_k^{(s)}(b_{i,k}^{(s)} + 1)) \\ &\quad + \sum_k \text{smin}(m_{i,k}^{(b)}(b_{i,k}^{(b)}), x_k^{(b)}(b_{i,k}^{(b)}), x_k^{(b)}(b_{i,k}^{(b)} + 1)), \end{aligned} \quad (19)$$

and

$$\begin{aligned} \max(x_i^{(s)}) &= c_i(\{b_{i,1}^{(s)}, \dots, b_{i,i-1}^{(s)}\}, \{b_{i,k}^{(b)}\}) + \sum_{k < i} \text{smax}(m_{i,k}^{(s)}(b_{i,k}^{(s)}), x_k^{(s)}(b_{i,k}^{(s)}), x_k^{(s)}(b_{i,k}^{(s)} + 1)) \\ &\quad + \sum_k \text{smax}(m_{i,k}^{(b)}(b_{i,k}^{(b)}), x_k^{(b)}(b_{i,k}^{(b)}), x_k^{(b)}(b_{i,k}^{(b)} + 1)) \end{aligned} \quad (20)$$

respectively. The above equations imply that we can fully translate a complete specification of buyer variable breakpoint intervals to a corresponding set of feasible supplier variable breakpoint intervals. We will search over the breakpoints using a branch-and-bound algorithm, and will need to be able to infer the restriction on the values of feasible supplier breakpoint intervals given a partial specification of buyer breakpoint interval variables. This inference, which narrows the domains of the supplier breakpoint variables, is called constraint propagation. Before describing how the branch-and-bound algorithm search over breakpoint interval variables is accomplished, we show how the constraint propagation may be obtained.

Narrowing of Supplier Breakpoint Domains Through Propagation

Let $\beta_k^{(b)}$ be a vector of integers specifying a set of breakpoint intervals for the k th buyer variable. $|\beta_k| = 1$ if the interval is uniquely determined, and $|\beta_k| = \kappa_k^{(b)} - 1$ if no restrictions on the interval have been specified. We index the l th component of $\beta_k^{(b)}$ as $\beta_{k,l}^{(b)}$, $l \in [1, \kappa_k^{(b)} - 1]$. A set of breakpoints for all buyer dimensions is written as $\{\beta_k^{(b)}\}$. In analogous fashion, we define $\beta_k^{(s)}$, $\beta_{k,l}^{(s)}$, $l \in [1, \kappa_k^{(s)} - 1]$, and $\{\beta_k\}$ for the supplier breakpoint interval variables.

From a specification $\{\beta_k^{(b)}\}$ we infer the feasible domains $\{\beta_k^{(s)}\}$ using $\max(x_i^{(s)})$ and $\min(x_i^{(s)})$ where:

$$\begin{aligned} \max(x_i^{(s)}) &= \sum_k \max_{\beta_k^{(b)} \in \beta_k^{(s)}} \left\{ c_{i,k}(\{b_k^{(b)}\}) + \text{smax}(m_{i,k}^{(b)}(b_k^{(b)}), x_k^{(b)}(b_k^{(b)}), x_k^{(b)}(b_k^{(b)} + 1)) \right\} + \\ &\quad \sum_{k < i} \max_{\beta_k^{(s)} \in \beta_k^{(s)}} \left\{ c_{i,k}(\{b_k^{(s)}\}) + \text{smax}(m_{i,k}^{(s)}(b_k^{(s)}), x_k^{(s)}(b_k^{(s)}), x_k^{(s)}(b_k^{(s)} + 1)) \right\} \end{aligned} \quad (21)$$

and

$$\min(x_i^{(s)}) = \sum_k \min_{b_k^{(b)} \in \beta_k^{(b)}} \left\{ c_{i,k}(\{b_k^{(b)}\}) + \text{smin}(m_{i,k}^{(b)}(b_k^{(b)}), x_k^{(b)}(b_k^{(b)}), x_k^{(b)}(b_k^{(b)} + 1)) \right\} + \sum_{k < i} \min_{b_k^{(s)} \in \beta_k^{(s)}} \left\{ c_{i,k}(\{b_k^{(s)}\}) + \text{smin}(m_{i,k}^{(s)}(b_k^{(s)}), x_k^{(s)}(b_k^{(s)}), x_k^{(s)}(b_k^{(s)} + 1)) \right\} \quad (22)$$

We begin with $i = 1$ (where no supplier variables contribute to $x_1^{(s)}$) and determine the interval $I_1 = [\min(x_1^{(s)}), \max(x_1^{(s)})]$. We then form the vector $\beta_1^{(s)}$ with components $\{\beta_{1,i}^{(s)} | [x_1^{(s)}(\beta_{1,i}), x_1^{(s)}(\beta_{1,i} + 1)] \cap I_1 \neq \emptyset\}$. Equations (21) and (22) can then be iterated to find the interval $I_2 = [\min(x_2^{(s)}), \max(x_2^{(s)})]$ using the previously calculated $\min(x_1^{(s)})$ and $\max(x_1^{(s)})$. With I_2 known, the components of $\beta_2^{(s)}$ are determined as above. Iterating this procedure, we can find the potentially narrowed domains of all $\{\beta_k^{(s)}\}$.

We now return to the optimization problem: we need to search over all $\prod_k \kappa_k^{(b)} \times \prod_k \kappa_k^{(s)}$ combinations of breakpoints to determine the combination that minimizes $d_1(x_{\text{opt}}(x, \{b_k^{(b)}\}, \{b_k^{(s)}\}), x)$. The simplest manner in which this accomplished is via exhaustive enumeration over all $\{b_k^{(b)}\}$ and $\{b_k^{(s)}\}$. However, once all $b_k^{(b)}$ have been specified, we need only search exhaustively over the corresponding $\{\beta_k^{(s)}\}$ inferred as above. There are, however, many ways to speed up the search over the buyer breakpoint intervals. We describe two methods.

D.3.2 Local Search

This method is extremely fast but not guaranteed to return the global optimum across all buyer breakpoint intervals. Here, initial random and feasible¹⁹ $\{b_k^{(s)}\}$, and $\{b_k^{(b)}\}$ sets are first created. After the continuous optimization is carried out using the initial $\{b_k^{(s)}\}$, and $\{b_k^{(b)}\}$, a new $\{\tilde{b}_k^{(s)}\}$, and $\{\tilde{b}_k^{(b)}\}$ set is generated from the old one by examining whether each generated $x_k^{(b)}(\{b_k^{(s)}\}, \{b_k^{(b)}\})$ solution is on the boundaries of the interval $x_k^{(b)}(b_k) \leq x_k^{(b)} \leq x_k^{(b)}(b_k + 1)$ or $x_k^{(s)}(b_k) \leq x_k^{(s)} \leq x_k^{(s)}(b_k + 1)$. If $x_k^{(b)}$ is on the lower bound of the interval, $\tilde{b}_k^{(b)} = b_k^{(b)} - 1$; similarly, if it is on the upper bound, $\tilde{b}_k^{(b)} = b_k^{(b)} + 1$. If $x_k^{(b)}$ is not on either end, then $\tilde{b}_k^{(b)} = b_k^{(b)}$. Of course this done while maintaining $\tilde{b}_k^{(b)}$ in the interval $1 \leq \tilde{b}_k^{(b)} \leq \kappa_k^{(b)} - 1$. Supplier breakpoint variables are updated in a similar manner. The procedure is repeated until no decrease in distance is attained.

D.3.3 Branch-and-Bound Search

Branch-and-bound is an efficient procedure used to exhaustively search a space and eliminate configurations that can be proven to be suboptimal. The theory behind branch-and-bound is well understood [5]. The idea is to structure the space as a tree of partially-completed possibilities and to explore the tree so that good complete solutions are constructed early on. Branches of the tree that can be shown to yield solutions

¹⁹Feasible in the sense that they do not violate the constraints in equations (10), (11), (12), and (13).

which cannot be better than the currently best solution need not be explored. Heuristics indicate the order in which to explore the tree and provide a lower bound on the fitness of partially completed solutions for pruning of suboptimal branches..

In the present context, we carry out branch-and-bound search over the space of possible buyer breakpoint combinations. The tree is structured beginning from a node at which none of the $b_k^{(b)}$ have been assigned. From this node, the tree branches into the $\kappa_i^{(b)} - 1$ possible assignments for $b_i^{(b)}$. From any of these nodes, the tree branches into the $\kappa_j^{(b)}$ possible values for $b_j^{(b)}$. The tree branches at each node to assign the remaining variables. At the leaves of the tree, all buyer breakpoint variables have been assigned.

From any node, the search proceeds by choosing the most promising node to consider, as indicated by a variable ordering heuristic and considering the possible values for that variable in the order indicated by the value ordering heuristic. A complete solution and its accompanying distance is obtained when the search ends at a leaf of the tree. To avoid unnecessary exploration, a lower bound heuristic indicates when a partially completed solution (node) need not be expanded because all completions will have higher distance than the best located so far. We describe these three heuristics for search over buyer breakpoints.

Lower Bound

Given a set of buyer breakpoint interval vectors $\{\beta_k^{(b)}\}$, we can propagate to $\{\beta^{(s)}\}$ and the intervals $[\min(x_i^{(s)}), \max(x_i^{(s)})]$. A lower bound on the value d_1 can be obtained by ignoring the supplier capabilities and minimizing

$$d_1(x) = (x^{(s)} - \mu)^t C_w^{-1} (x^{(s)} - \mu) \quad (23)$$

subject to

$$G_1 x^{(s)} = g_1 \quad (24)$$

$$G_2 x^{(s)} \leq g_2 \quad (25)$$

$$\min(x_i^{(s)}) \leq x_i^{(s)} \leq \max(x_i^{(s)}), \quad (26)$$

The lower bound thus obtained is denoted by $lb(\{\beta^{(b)}\})$.

Variable and Value Ordering

The lower bound $lb(\{\beta^{(b)}\})$ can be used to define an ordering over variables and the values any variable may take on. We define the functions $L_{k,l}$ which operate on the set $\{\beta^{(b)}\}$ by locking the k th buyer breakpoint variable to $\beta_{k,l}$. At any node we can determine the lower bounds $lb(L_{k,l}\{\beta^{(b)}\})$. We can select to explore the variables in order of increasing lb_k where

$$lb_k(\{\beta^{(b)}\}) = \frac{\sum_{l=1}^{|\beta_k^{(b)}|} lb(L_{k,l}(\{\beta^{(b)}\}))}{\sum_{l=1}^{|\beta_k^{(b)}|} 1},$$

i.e., the average over all possible completions for $b_k^{(b)}$.²⁰ Within any given variable the (say k^*) values can be explored in order of increasing $\text{lb}(L_{k^*,i}(\{\beta^{(b)}\}))$. If we are willing to do more computation, we can explore the branches in an interleaved order according to all the $\text{lb}(L_{k,i}(\{\beta^{(b)}\}))$.

Pruning and Search

During the branch-and-bound search, branches can be pruned if the node has $\text{lb}(L_{k,i}(\{\beta^{(b)}\}))$ greater than the distance of the lowest examined leaf node.

Search begins at the root node where each $\beta_k^{(b)} = [1, \dots, \kappa_k^{(b)} - 1]$. We applying the ordering heuristics above to move to a node where one breakpoint variable has been fixed. From that node, we move down the tree, fixing a value at a time until we reach a leaf. We then progressively backtrack up the tree, eliminating branches whose lower bounds are greater than the best leaf examined.

Once the optimal set of buyer breakpoints has been determined, we search exhaustively over the supplier breakpoints feasible with this choice according to Equations (19) and (20).

Improvements

We have not yet exploited the fact that, from any node having $\{\beta_k^{(b)}\}$, the constraints in Equation (12) may narrow the domains of the buyer breakpoint variables deeper in the tree. The computational cost of narrowing the breakpoint domains may be warranted if the pruning thus obtained is sufficient.

E Phase 2 Optimization

In this section we detail an efficient algorithm to optimize Eq. (5):

$$\arg \min_{\mathbf{x}} d_1(\mathbf{x}(\mathbf{x}), \mathbf{x}) + d_2(\mathbf{x}) \quad \text{subject to} \quad C^{(b)}(\mathbf{x}) \wedge C^{(s)}(\mathbf{x}).$$

Our approach is a marriage of constraint programming and local search.

E.1 Local Search over D -neighborhoods

We begin with a definition. The d -neighbors of a discrete configuration \mathbf{x} are all those configurations, \mathbf{x}' , which differ from \mathbf{x} in d attributes. Attribute i differs in two configurations if $x_i \neq x'_i$. Thus the distance varies from 0 if the configurations are identical, up to n_d if all attributes are different. The D -neighborhood of \mathbf{x} is the union of all $d = 1$ neighbors, $d = 2$ neighbors, up to $d = D$ neighbors. The D -neighborhood around \mathbf{x} is indicated by $\mathcal{N}_D(\mathbf{x})$. We note that, for any configuration, its n_d neighborhood is the entire discrete space.

Using the D -neighborhood, we apply a standard hillclimbing algorithm which respects all the discrete constraints. D is specified before the algorithm begins and serves as an efficient manner to trade off running

²⁰We might also take the minimum over all possible completions $\text{lb}_k(\{\beta^{(b)}\}) = \min_{i \in [1, |\beta_k^{(b)}|]} \text{lb}(L_{k,i}(\{\beta^{(b)}\}))$.

1	***11	1	***+1
2	**1*1	2	**1*1
3	**11*	3	**1+*
4	*1**1	4	*+**1
5	*1*1*	5	*+**+
6	*11**	6	*+1**
7	1***1	7	1***1
8	1**1*	8	1**+*
9	1*1**	9	1*1**
10	11***	10	1+***

(a)
(b)

Table 5: Configurations examined in the $D = 3$ neighborhood search around (a) the initial configuration 11111 and (b) the initial configuration 0+0+0 (duplicate search configurations are indicated in bold).

time with the quality of the final solution. The hillclimbing algorithm takes as input a starting configuration \mathbf{x}_t and finds the configuration $\mathbf{x}_{t+1} \in \mathcal{N}_D(\mathbf{x}_t)$ which minimizes $d_1(\mathbf{x}(\mathbf{x}), \mathbf{x}) + d_2(\mathbf{x})$ subject to $\mathcal{C}^{(b)}(\mathbf{x}) \wedge \mathcal{C}^{(c)}(\mathbf{x})$. The algorithm then iterates again starting from \mathbf{x}_{t+1} . The hillclimbing algorithm terminates when it finds a configuration \mathbf{x}_T which has a distance less than or equal to all of its D -neighbors. If $D = n_d$, this is the global minimum. Given a function which exhaustively searches a D -neighborhood for the lowest distance feasible configuration, a D -neighborhood hillclimbing algorithm is easily implemented.

There is one noteworthy point about the implementation of an efficient D -neighborhood hillclimber. Consider a $D = 3$ search over $n_d = 5$ discrete variables. Without loss of generality, we label the initial configuration as 11111. In the $D = 3$ neighborhood search, the configurations examined are given in Table 5(a) as indicated by the *s. Let us assume that the best neighbor in the D neighborhood of 11111 differs in components 2 and 4. We indicate this best neighbor as 1+1+1, which serves as the starting configuration for a new $D = 3$ search. The configurations examined in the second neighborhood search are given in Table 5(b). From Table 5(b) we see that the search branches 2, 7, and 10 are redundant, having already been performed in the D neighborhood search around 11111.²¹ Thus these branches may be pruned in the second D neighborhood search. A search branch covers a previous search and may be pruned if the vector of positions of the wildcard characters covers the vector of positions at which the previous best solution was found. The binary relation covers is defined as

$$\mathbf{x} \text{ covers } \mathbf{x}' \iff \mathbf{x}'_i \in \mathbf{x} \forall \mathbf{x}'_i \in \mathbf{x}'.$$

²¹The duplication of search effort is most acute when D is large and the fittest neighbor is found very close by (small d).

The covers relation is transitive but not reflexive. Thus, to increase the efficiency, we can keep a list of the previous branches searched and ignored search branches that have already been covered.

E.1.1 Improvements

Throughout the present discussion we have assumed that D remains constant through the hillclimbing process. There are good reasons why this should not be the case. With any reasonable landscape correlation structure, we know that lower distance neighbors are more likely to be found early in the hillclimbing process. Thus a reasonable heuristic is to start with a larger D early on and gradually lower it with time.

E.2 Exhaustive Search of a D -neighborhood

To complete the discussion of the phase 2 optimization, we describe a computationally efficient way in which to find the lowest distance feasible configuration within a D -neighborhood. We use a branch-and-bound method with heuristics tuned for the present problem. The factors tuned include the variable order in which we search, the order of the values any given variable may take, and a heuristic which says when certain branches of the search space may be safely ignored.

We use heuristics which early on in the search generate low-distance configurations, and later prune parts of the search that we can prove must have higher distance than the best configuration generated so far.

E.2.1 Heuristics And Lower Bounds

We focus on obtaining good heuristics for the discrete contribution to the total distance, then later incorporate the continuous and range contributions since these are simpler.

At any point in the search, some variables have been assigned and some have not. We divide up the sum over the $Z_{i,j}(\mathbf{x}_i, \mathbf{x}_j)$ into four contributions

$$\begin{aligned}
 Q_d Z_w(\mathbf{x}) &= Q_d \sum_{i \text{ locked}} \sum_{j \text{ locked}} w_i w_j Z_{i,j}(\mathbf{x}_i, \mathbf{x}_j) + Q_d \sum_{i \text{ locked}} \sum_{j \text{ floating}} w_i w_j Z_{i,j}(\mathbf{x}_i, \mathbf{x}_j) \\
 &\quad + Q_d \sum_{i \text{ floating}} \sum_{j \text{ locked}} w_i w_j Z_{i,j}(\mathbf{x}_i, \mathbf{x}_j) + Q_d \sum_{i \text{ floating}} \sum_{j \text{ floating}} w_i w_j Z_{i,j}(\mathbf{x}_i, \mathbf{x}_j) \\
 &\equiv Z_{l,l} + Z_{l,f} + Z_{f,l} + Z_{f,f}
 \end{aligned}$$

Since we are minimizing and all $Z_{i,j}(\mathbf{x}_i, \mathbf{x}_j)$, w_i , and Q_d are positive, we can prune any search node which has $Z_{l,l} >$ than the distance of the lowest distance configuration found thus far, since additional terms will only add to the sum.

We obtain a lower bound on the distance by lower-bounding each of the above four terms. Just taking the lowest value in the tables can yield differing values for any given variable, e.g., the values for a single

variable x_i :

$$x_i^* = \arg \min_{x_i} Z_{i,j}(x_i, x_j), \quad x_i^* = \arg \min_{x_i} Z_{j,i}(x_j, x_i), \quad \text{and} \quad x_i^* = \arg \min_{x_i} Z_{i,i}(x_i)$$

may all be different. To correct this we write

$$Q_d Z_w = Z_{l,l} + Z_{f,f} + \sum_{i \text{ floating}} Z_{l,f}^{(i)}(x_i^*)$$

where $Z_{l,l}$ and $Z_{f,f}$ are calculated as before and

$$Z_{l,f}^{(i)}(x_i) = Q_d \left(w_i^2 Z_{i,i}(x_i) + w_i \sum_{j \text{ locked}} w_j (Z_{i,j}(x_i, x_j) + Z_{j,i}(x_j, x_i)) \right).$$

The best choice for the floating variable x_i is

$$x_i^* = \arg \min_{x_i} Z_{l,f}^{(i)}(x_i). \quad (27)$$

In this way all the possible choices for x_i are at least consistent, and the true lower limit is still guaranteed to be higher than this (since the $Z_{f,f}$ contribution can only be higher).

This approach also naturally suggests a variable ordering and a value ordering within any variable. We can try the variables in increasing order of their corresponding $Z_{l,f}^{(i)}(x_i^*)$ so that variable i , which has the smallest $Z_{l,f}^{(i)}(x_i^*)$, is tried first. For any given variable x_i we can also order the possible value assignments for x_i according to the array of values $Z_{l,f}^{(i)}(x_i)$ so that the value $x_i = x_i^*$ is tried first and the value which maximizes $Z_{l,f}^{(i)}(x_i)$ is tried last.

With little extra work we can impose a consistency requirement raising the lower bound for $Z_{f,f}$. We decompose the sum as follows

$$Z_{f,f} \equiv Q_d \sum_{i,j \text{ floating}; j \neq i} w_i w_j Z_{i,j}(x_i, x_j) = Q_d \sum_{i,j \text{ floating}; j < i} w_i w_j (Z_{i,j}(x_i, x_j) + Z_{j,i}(x_j, x_i)).$$

Instead of taking the minimum component-wise as previously suggested, i.e. instead of

$$Z_{f,f}^* = Q_d \sum_{i,j \text{ floating}; j \neq i} \min_{x_i, x_j} [w_i w_j Z_{i,j}(x_i, x_j)],$$

we take the minimum so that

$$Z_{f,f}^* = Q_d \sum_{i,j \text{ floating}; j < i} \min [w_i w_j (Z_{i,j}(x_i, x_j) + Z_{j,i}(x_j, x_i))]. \quad (28)$$

which enforces pairwise consistency between variables i and j .

The lower bound, lb, we have obtained is thus

$$\text{lb} = Z_{l,l} + Z_{f,f}^* + \sum_i Z_{l,f}^{(i)}(x_i^*)$$

where x_i^* is determined from Eq. (27) and $Z_{i,f}^*$ is determined from Eq. (28).

Integration Including Continuous and Range Contributions

In this section we summarize the complete heuristic specification, including the effects of the continuous and range variables. As above, we use the definitions

$$\begin{aligned} Z_{l,1} &= Q_d \sum_{i,j \text{ locked}} w_i w_j Z_{i,j}(x_i, x_j) \\ Z_{l,f}^{(i)}(x_i) &= Q_d w_i \sum_{j \text{ locked}} w_j (Z_{i,j}(x_i, x_j) + Z_{j,i}(x_j, x_i)) \\ Z_{i,f}^* &= Q_d \sum_{i,j \text{ floating}; j < i} \min [w_i w_j (Z_{i,j}(x_i, x_j) + Z_{j,i}(x_i, x_j))] \end{aligned}$$

so that a lower bound to the complete discrete contribution to the distance is $Z_{l,1} + Z_{i,f}^* + \sum_i Z_{l,f}^*(x_i^*)$ where $x_i^* = \arg \min_{x_i} Z_{l,f}^{(i)}(x_i)$.

We now extend this result to include the continuous and range variables. The continuous and range variables depend on the discrete variables through the mask. Since we expect there will be relatively few masks we exhaustively enumerate all the discrete possibilities. Consequently, we define $d_1(x_{\text{floating}} | x_{\text{locked}})$ as the distance contribution made by the continuous and range variables (the d_1 contribution to Eq. (5)). We define the continuous + range contribution for any given floating variable i as

$$d_1^{(i)}(x_i) \equiv \min_{\text{floating} \setminus i} d_1(x_{\text{floating}} | x_{\text{locked}})$$

where the min is taken over all floating variables other than i and where floating variable i is locked to value x_i .

Since each continuous/range/discrete distance contributes additively to the total distance, we can rank the possible values for any given floating variable i according to $Z_{l,f}^{(i)}(x_i) + d_1^{(i)}(x_i)$. The best (according to this heuristic) setting for floating variable i is

$$x_i^* = \arg \min_{x_i} \{ Z_{l,f}^{(i)}(x_i) + d_1^{(i)}(x_i) \}.$$

The heuristic then suggests a variable ordering in order of increasing $Z_{l,f}^{(i)}(x_i^*) + d_1^{(i)}(x_i^*)$ so that we first select the floating variable with the lowest sum and last select the variable with the highest sum.

Finally, we turn to the lower bound on the distance. Noting that the continuous and range distance contribution does not have the form of a sum over the dimensions (as the discrete contribution does), we write the lower bound, lb, as²²

$$\text{lb} = Z_{l,1} + \sum_i (Z_{l,f}^{(i)}(x_i^*) + \delta_{i,i^*} d_1^{(i)}(x_i^*)) + Z_{i,f}^*$$

where $i^* = \arg \min_i (Z_{l,f}^{(i)}(x_i^*) + d_1^{(i)}(x_i^*))$ (i.e., i^* labels the floating dimension with the least $Z_{l,f}^{(i)} + d_1^{(i)}$ contribution to the distance).

²² $\delta_{x,x'}$ is the Kronecker delta function which is 1 if and only if $x = x'$.

E.2.2 Improvements

We point out that an extensive literature exists on parallel branch and bound algorithms. Many of the techniques developed are applicable to the present problem.

AMENDED CLAIMS

[received by the International Bureau on 8 May 2001 (08.05.01);
new claims 3-108 added; remaining claims unchanged (14 pages)]

1. A method for discovery of trades between one or more buyers and one or more sellers comprising the steps of:
 - 5 (a) expressing one or more terms of an ideal trade and one or more flexibilities by at least one of said buyers;
 - (b) expressing one or more capabilities by at least one of said sellers; and
 - (c) determining at least one optimal trade with respect to said one or more terms and said one or more flexibilities of said at least one buyer and said one or more capabilities of said at least one seller.
- 10 2. A method for discovery of trades between one or more buyers and one or more sellers as in 1, wherein said one or more terms comprise one or more members of the group consisting of continuous factors, discrete factors and range factors.
- 15 3. (New) A system for determining one or more trades between a buyer and one or more suppliers comprising:
 - (a) one or more variables defining a space of negotiation;
 - (b) a utility function of said one or more variables for expressing a utility of the one or more trades to the buyer over the space of negotiation comprising:
 - 20 i. an ideal trade to the buyer defined by one or more ideal values corresponding to said one or more variables; and
 - ii. at least one flexibility in at least one of said variables expressing how the utility of the trade to the buyer varies in the space of negotiation for said ideal trade;
 - (c) one or more capabilities for defining a subspace of the negotiations space wherein the one or more suppliers have an ability to trade; and
 - 25 (d) an optimizer determining at least one of the trades that is optimal with respect to said utility of the buyer subject to the capabilities of the one or more suppliers.
4. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 3 wherein said one or more variables comprise one or more of the following:
 - 30 (a) one or more continuous variables x , one or more discrete variables κ and one or more range variables r .
5. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 4 wherein each x_i of said one or more continuous variables x has an allowed range over which said each continuous variable x_i may vary $x_i \in X_i = [\underline{x}_i, \bar{x}_i]$, wherein \underline{x}_i is a lower bound of said continuous variable x_i and \bar{x}_i is an upper bound of said continuous variable x_i ;
- 35

6. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 5 wherein each κ_i of said one or more discrete variables κ has a value from a domain
- 5 $\kappa_i \in \mathcal{D}_i = [1, \dots, d_i]$ where $d_i \geq 0$ is an integer giving the number of possible values that said discrete variable κ_i may assume.
7. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 6 wherein the space of negotiation comprises a tensor product $X_1 \otimes \dots \otimes X_{n_c} \otimes \mathcal{D}_1 \otimes \dots \otimes \mathcal{D}_{n_d}$.
- 10 wherein
- n_c is the number of said continuous variables;
- n_d is the number of said discrete variables;
- X_i is said allowed range of said continuous variable x_i ; and
- \mathcal{D}_i is said domain of said discrete variable κ_i .
8. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 4 wherein said utility function $u((x, \kappa, r))$ comprises an expression of a distance function $d(x, \kappa, r)$ that defines a distance from said ideal trade in the space of negotiation.
- 15
9. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 8 wherein said distance function comprises at least one of the following: a continuous distance, a discrete distance $Z(\kappa)$, and a range distance R .
- 20
10. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 9 wherein said range distance depends on the value of at least one of said discrete variables κ .
11. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 10 wherein said range distance is defined as:
- 25
$$R(r; \kappa) = \sum_{i=1}^{n_r} R_i(r_i; \kappa)$$
- wherein:
- r is an n_r vector of tables of preferred values of said one or more range variables, $r_i = (\underline{r}_i, \bar{r}_i)$;
- 30 n_r is the number of said range variables; and
- $\bar{r}_i > \underline{r}_i$.
12. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 10 wherein said range distance is a distance $d(r_i, r_j)$ between said range variable $r_i = (\underline{r}_i, \bar{r}_i)$ of the buyer and said range variable $r_j = (\underline{r}_j, \bar{r}_j)$ of at least
- 35 one of the suppliers.

13. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 12 wherein said distance between said buyer range variables and said supplier range variable $d(r_i, r_j)$ comprises an overlap between said buyer range variable and said supplier range variables, overlap (r_i, r_j) .

5 14. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 13, wherein said overlap is defined as

$$\text{overlap}(r_i, r_j) = \int_{\mu_j - \sigma_j}^{\mu_j + \sigma_j} dx \mathcal{N}(x; r_i) \mathcal{N}(x; r_j)$$

where

10 $\mathcal{N}(x; r_j)$ is a Gaussian in x centered at $\mu_j = (\underline{r}_j + \bar{r}_j)/2$ with standard deviation $\sigma_j = \alpha(\bar{r}_j - \underline{r}_j)$;

$\mathcal{N}(x; r_i)$ is a Gaussian in x centered at $\mu_i = (\underline{r}_i + \bar{r}_i)/2$ with standard deviation $\sigma_i = \alpha(\bar{r}_i - \underline{r}_i)$; and

α is a tunable parameter.

15 15. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 14 wherein said range distance is defined as

$$R(r_i, r_j) = -\ln \left[\frac{\text{overlap}(r_i, r_j)}{\text{maxOverlap}} \right]$$

wherein

$$\text{maxOverlap} = \frac{\text{erf} \left[\frac{1}{\sqrt{2}} \right]}{\sqrt{2\pi(\sigma_i^2 + \sigma_j^2)}}$$

20

16. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 9 wherein said continuous distance is quadratic and is determined by a positive semi definite $n_c \times n_c$ matrix C^{-1} wherein n_c is the number of said continuous variables.

25 17. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 16 wherein said continuous distance is defined as

$$(x - \mu(\kappa))^t C^{-1}(\kappa) (x - \mu(\kappa)).$$

wherein

30 μ is an n_c -vector of ideal values of said continuous variables that may depend on said discrete variable κ .

18. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 9 wherein said continuous distance is a convex function.
19. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 9 wherein each of said discrete variables x_i has a value from a domain D_i .
20. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 19 wherein said discrete distance $Z(x)$ maps a discrete space $D_1 \otimes \dots \otimes D_{n_d}$ onto the positive real line $[0, \infty]$ wherein n_d is the number of said discrete variables.
21. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 20 wherein said discrete distance $Z(x)$ is defined as

$$Z(x) = \sum_{i=1}^{n_d} \left\{ Z_i(x_i) + \sum_{j=1(\neq i)}^{n_d} Z_{i,j}(x_i, x_j) \right\}$$

wherein

- each $Z_{i,j}$ is a table comprising $d_i d_j$ entries; and
- $d_i d_j$ is the distance if said i th discrete variable has value x_i , conditional on said j th discrete variable having value x_j .
22. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 21 wherein values in said tables $Z_{i,j}$ are determined from one or more rankings by the buyer.
23. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 22 wherein said discrete distance Z is defined as $Z = -\ln[1 - Z']$.
- wherein
- Z' is normalized to lie between 0 and 1.
24. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 23 wherein said normalized distance Z' is a linear scaling.
25. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 23 wherein said normalized distance Z' is an exponential scaling.
26. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 23 wherein said normalized distance Z' is an algebraic scaling.
27. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 9 wherein said discrete distance $Z(x)$ is a function of one or more pairs of said discrete variables x_i, x_j .

28. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 8 wherein said distance is generated from a ranking of preferred values for said one or more variables by the buyer.
29. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 4 wherein said utility function $u((x, \kappa, r))$ expresses one or more tradeoffs among the one or more variables x, κ, r .
30. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 8 further comprising one or more scaling factors, Q_c, Q_d, Q_r to normalize contributions of said at least one continuous variable x , said at least one discrete variable κ , and said at least one range variable r to said utility function $u((x, \kappa, r))$ for defining a baseline wherein the one or more variables contribute equally to said utility.
31. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 30 wherein said distance function with said normalized contribution is defined as:

$$d(x, \kappa, r) = d_c + Q_d d_d + Q_r d_r$$

wherein

d_c is a contribution to said distance by said continuous variables,

$Q_d d_d$ is a contribution to said distance by said discrete variables, and

$Q_r d_r$ is a contribution to said distance by said range variables.

32. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 30 wherein values of said scaling factors are set so that average distances of said one or more from variables from said corresponding one or more ideal values are equal.
33. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 32 wherein said average distance comprises utility-weighted average distances over said space of negotiation for placing more weight on better ones of the trades.
34. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 33 wherein said utility-weighted average distances are defined as

$$\begin{aligned} (d_r) &\equiv \frac{\sum_{\kappa} \sum_r \int_V du Q_r d_r \exp[-Q_r d_r - Q_d d_d - d_c]}{\sum_{\kappa} \sum_r \int_V du \exp[-Q_r d_r - Q_d d_d - d_c]} = Q_r \frac{\sum_{\kappa} \exp[-Q_d d_d] \sum_r d_r \exp[-Q_r d_r] \int_V du \exp[-d_c]}{\sum_{\kappa} \exp[-Q_d d_d] \sum_r \exp[-Q_r d_r] \int_V du \exp[-d_c]} \\ (d_d) &\equiv \frac{\sum_{\kappa} \sum_r \int_V du Q_d d_d \exp[-Q_r d_r - Q_d d_d - d_c]}{\sum_{\kappa} \sum_r \int_V du \exp[-Q_r d_r - Q_d d_d - d_c]} = Q_d \frac{\sum_{\kappa} d_d \exp[-Q_d d_d] \sum_r \exp[-Q_r d_r] \int_V du \exp[-d_c]}{\sum_{\kappa} \exp[-Q_d d_d] \sum_r \exp[-Q_r d_r] \int_V du \exp[-d_c]} \\ (d_c) &\equiv \frac{\sum_{\kappa} \sum_r \int_V du d_c \exp[-Q_r d_r - Q_d d_d - d_c]}{\sum_{\kappa} \sum_r \int_V du \exp[-Q_r d_r - Q_d d_d - d_c]} = \frac{\sum_{\kappa} \exp[-Q_d d_d] \sum_r \exp[-Q_r d_r] \int_V du d_c \exp[-d_c]}{\sum_{\kappa} \exp[-Q_d d_d] \sum_r \exp[-Q_r d_r] \int_V du \exp[-d_c]} \end{aligned}$$

wherein

- \sum_{κ} indicates a repeated sum $\sum_{\kappa_1} \cdots \sum_{\kappa_n}$ over all possible discrete trades;
- \sum_r indicates a sum over all of said range variables; and

$Q_c = 1$ because Q_r is interpreted as Q_r/Q_c and Q_d is interpreted as Q_d/Q_c .

35. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 34 wherein said scaling factors Q_c , Q_d , Q_r are determined from said utility-weighted average distances.

5 36. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 30 further comprising one or more weights to enable the buyer to weight said contributions of said one or more variables to said utility.

10 37. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 36 wherein said distance function comprises at least one of the following: a weighted continuous distance, a weighted discrete distance $Z_w(\mathbf{x})$ and a weighted range distance.

38. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 37 wherein said weighted continuous distance is defined as:

15
$$(\mathbf{x} - \boldsymbol{\mu}(\mathbf{x}))^t \mathbf{C}_w^{-1}(\mathbf{x}) (\mathbf{x} - \boldsymbol{\mu}(\mathbf{x})).$$

wherein

$$\mathbf{C}_w^{-1} = \mathbf{W}_c \mathbf{C}^{-1} \mathbf{W}_c;$$

\mathbf{W}_c is a diagonal matrix formed from \mathbf{w}_c ;

\mathbf{w}_c is an n_c -vector of weight for said continuous variables; and

20 $\boldsymbol{\mu}$ is an n_c -vector of ideal values for said continuous variables that may depend on said discrete variables \mathbf{x} .

39. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 37 wherein said weighted discrete distance $Z_w(\mathbf{x})$ is defined as:

$$Z_w(\mathbf{x}) = \sum_{i=1}^{n_d} w_{d,i} \{w_{d,i} Z_i(\mathbf{x}_i) + \sum_{j=1(\neq i)}^{n_d} w_{d,j} Z_{i,j}(\mathbf{x}_i, \mathbf{x}_j)\}$$

25 wherein

$Z_{i,j}(\mathbf{x}_i, \mathbf{x}_j)$ is the distance if said i th discrete variable has value \mathbf{x}_i conditioned on said j th discrete variable having value \mathbf{x}_j ; and

$w_{d,i}$ is the i th component of a n_d vector of weights for said discrete variables.

30 40. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 37 wherein said weighted range distance is defined as

$$R_w(\mathbf{r}) = \sum_{i=1}^{n_r} w_{r,i} R_i(\mathbf{r}_i)$$

wherein n_r is the number of said range variables,

\mathbf{r} is an n_r -vector of tuples of preferred values of said one or more range variables,

$\mathbf{r}_i = (\underline{r}_i, \bar{r}_i)$; and

35 $\bar{r}_i > \underline{r}_i$.

41. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 37 wherein said weighted distance function comprises at least one of the following: an n_c -vector of weights for said continuous variables, w_c , an n_d -vector of weights for said discrete variables, and an n_r -vector of weights for said range variables.
- 5 42. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 41 wherein said weights are normalized so that
- $$C_w^{-1} = W_c C^{-1} W_c$$
43. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 41 wherein values for said weights depend on values of said discrete
- 10 variables, x_i .
44. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 4 further comprising a total cost of ownership function expressing a total cost of membership to the buyer that varies over the negotiation space.
45. (New) A system for determining one or more trades between a buyer and one or more
- 15 suppliers as in claim 44 wherein said total cost of ownership function comprises one or more cost contribution.
46. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 45 wherein said one or more cost contributions comprise one or more of the following: piece part costs, freight costs, setup costs, quality assurance
- 20 costs, repair costs, and revenue generated from the trade.
47. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 44 wherein said total cost of ownership function is defined as
- $$C_o(x, x, r; \beta)$$
- wherein
- 25 β represents one or more other factors comprising one or more of the following: forecasted demand and current inventory levels.
48. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 47 wherein said one or more other factors are extracted from at least one of the following: an enterprise resource planning system and a supply chain
- 30 management system.
49. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 48 wherein said one or more other factors are extracted in real time for enabling continuous, real time optimization.
50. (New) A system for determining one or more trades between a buyer and one or more sup-
- 35 pliers as in claim 47 wherein minimization of said cost of ownership function $C_o(x, x, r; \beta)$ determines said ideal trade to the buyer: $x_{opt}(\beta)$, $x_{opt}(\beta)$, $r_{opt}(\beta)$

51. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 50 wherein said at least one flexibility is determined by a Hessian matrix $\mathbf{H} = [h_{i,j}]$

wherein $h_{i,j}$ is defined as

5

$$h_{i,j} = \frac{\partial^2 C_o(\mathbf{x}, \boldsymbol{\kappa}, \mathbf{r}; \boldsymbol{\beta})}{\partial x_i \partial x_j} \bigg|_{\mathbf{x}=\mathbf{x}_{\text{opt}}(\boldsymbol{\beta}), \boldsymbol{\kappa}=\boldsymbol{\kappa}_{\text{opt}}(\boldsymbol{\beta}), \mathbf{r}=\mathbf{r}_{\text{opt}}(\boldsymbol{\beta})}$$

52. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 3 wherein said one or more capabilities comprise one or more of the following: price discounts on large volume orders, and variation in delivery as a function of price.

10

53. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 8 wherein said one or more capabilities specify one or more of the following:

one or more continuous capabilities, one or more discrete capabilities, and allowed values for said range variables.

15

54. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 53 wherein said allowed values for said ranges variables contribute to said distance function.

55. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 53 wherein said allowed values for said range variables comprise one or more pairs $(\underline{r}_j, \bar{r}_j)$ wherein \underline{r}_j is a lower bound for the j th one of said range variable and \bar{r}_j is an upper bound for the j th one of said range variable.

20

56. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 52 wherein said continuous capabilities are one or more responses from said suppliers to a request for the buyer.

25

57. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 56 further comprising a vector-valued function $\mathbf{f}(\mathbf{x}^{(b)}, \mathbf{x}^{(s)}, \boldsymbol{\kappa})$ to determine said one or more supplier responses wherein

$\mathbf{x}^{(b)}$ is said buyer request and

$\mathbf{x}^{(s)}$ is at least a previous one of said supplier responses.

30

58. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 57 wherein said vector-valued function $\mathbf{f}(\mathbf{x}^{(b)}, \mathbf{x}^{(s)}, \boldsymbol{\kappa})$ comprises one or more components f_i for corresponding ones of said continuous variables:

$$x_i^{(s)} = f_i(\mathbf{x}^{(b)}, \mathbf{x}^{(s)})$$

59. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 52 wherein said one or more components, f_i are piecewise linear functions.
- 5 60. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 59 wherein said one or more components, f_i are specified with one or more breakpoints.
- 10 61. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 53 wherein said one or more discrete capabilities of said suppliers comprise one or more constraints from said suppliers on said one or more discrete variables.
- 15 62. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 3 wherein said one or more capabilities are represented by one or more piecewise linear functions.
63. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 4 further comprising one or more constraints involving said one or more variables which must be satisfied for the buyer and at least one of the sellers to trade.
- 20 64. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 63 wherein said one or more constraints comprise one or more of the following: discrete constraints for expressing one or more allowed and/or disallowed combinations of values for said discrete variables x and continuous constraints for setting one or more requirements on said continuous variable x .
65. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 64 wherein said continuous constraints from the buyer are linear.
- 25 66. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 64 wherein said continuous constraints comprise at least one of inequality constraints and equality constraints.
- 30 67. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 64 wherein said continuous constraints depend on values of said discrete variables, x .
68. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 63 wherein said one or more constraints comprise one or more of the following:
required delivery time, and an unacceptable color.
- 35 69. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 64 wherein at least one of said continuous constraints depend on values of at least one of said discrete variables x .

determining values of said continuous variables that minimize said distance for one or more settings of said discrete variables.

79. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 78 further comprising the steps of:

5 representing said distance by a function of said discrete variables; and
determining an optimal one of said settings of said discrete variables by minimizing said function of said discrete variables.

80. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 78 wherein said determining values for said continuous variable step
10 is performed under one or more constraints on said continuous variables.

81. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 79 wherein said representing said determining an optimal one of said settings of said discrete variables step is performed under one or more constraints on said discrete variables.

15 82. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 4 further comprising means for aggregating at least one of the suppliers to participate in said trade with the buyer.

83. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 82 wherein said aggregating means perform the steps of:
20 determining one or more subsets of said suppliers that satisfy one or more constraints on said discrete variables.

84. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 83 wherein said one or more discrete variable constraints comprise at least one of the following: buyer discrete variable constraints and seller discrete variable constraints.
25

85. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 83 further comprising the step of optimizing over said continuous variables to determine an optimal one of said subset of buyers.

86. (New) A grammar for a system that determines one or more trades between a buyer and one or more suppliers comprising:
30

- (a) one or more capability rules representing one or more capabilities of said suppliers to trade with the buyer;
- (b) one or more preference rules representing one or more preferences of the buyer; and
- (c) one or more match rules representing matches between said one or more capabilities
35 and said one or more preferences.

70. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 3 wherein said determining at least one of the trades that is optimal step comprises the steps of:
- 5 selecting one of said suppliers;
- determining at least one of the trades that corresponds to a maximum value of said utility of the buyer and that is within the capabilities of said selected supplier.
71. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 70 wherein said determining at least one of the trades that is optimal step further comprises the steps of:
- 10 selecting another of said suppliers; and
- repeating said determining at least one of the trades step for said another selected supplier.
72. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 71 wherein said determining at least one of the trades that is optimal step further comprises the steps of:
- 15 choosing at least of the suppliers having the highest said maximum value of said utility of the buyer.
73. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 72 further comprising a subsystem to perform said determined trade between the buyer and the chosen supplier.
- 20
74. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 3 further comprising a subsystem to perform said determined trade.
75. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 3 wherein said utility comprises at least one of the following: quantitative factors and qualitative factors.
- 25
76. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 4 wherein said determining at least one of the trades that is optimal step further comprises the step of:
- minimizing a distance from said ideal trade.
- 30
77. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 76 wherein said distance comprises one or more of the following: a continuous component, a discrete component, and a range component.
78. (New) A system for determining one or more trades between a buyer and one or more suppliers as in claim 77 wherein said minimizing a distance from said ideal trade step comprises the steps of:
- 35

87. (New) A grammar for a system that determines one or more trades as in claim 86 wherein said one or more capabilities of said suppliers are specific to particular ones of said buyers.
- 5 88. (New) A grammar for a system that determines one or more trades as in claim 86 wherein said capability rules comprise one or more of the following:
- (a) a discrete variable rule for representing a description of a discrete variable;
 - (b) a continuous variable rule for representing a description of a continuous variable; and
 - (c) a range variable rule for representing a description of a range variable.
- 10 89. (New) A grammar for a system that determines one or more trades as in claim 88 wherein said description of the continuous variable comprises at least one of a minimum value and a maximum value for the continuous variable.
- 15 90. (New) A grammar for a system that determines one or more trades as in claim 88 wherein said one or more capability rules comprise one or more constraint rules representing constraints on value of at least one of said discrete variables, and said continuous variables.
91. (New) A grammar for a system that determines one or more trades as in claim 89 wherein said one or more constraint rules comprise at least one matrix for representing said constraints.
- 20 92. (New) A grammar for a system that determines one or more trades as in claim 90 wherein said constraints on said values of said discrete variables comprise one or more permitted value continuations for the discrete variables.
- 25 93. (New) A grammar for a system that determines one or more trades as in claim 88 wherein said range variable description comprises at least one of a minimum value and a maximum value for the range variable.
94. (New) A grammar for a system that determines one or more trades as in claim 90 wherein said constraints on said values of said continuous variables comprise one or more of the following: an inequality, an equality, a linear constraint, and a non-linear constraint.
- 30 95. (New) A grammar for a system that determines one or more trades as in claim 86 wherein said one or more capability rules further comprise an aggregation flag indicating a willingness of the supplier to participate in an aggregation for the buyer.
96. (New) A grammar for a system that determines one or more trades as in claim 86 wherein said preferences of the buyer are specific to at least one of said suppliers.
- 35 97. (New) A grammar for a system that determines one or more trades as in claim 86 wherein said preference rules comprise one or more of the following:

- (a) a continuous variable rule for representing a description of a continuous variable;
 - (b) a discrete variable rule for representing a description of a discrete variable, and
 - (c) a range variable rule for representing a description of a range variable.
- 5 98. (New) A grammar for a system that determines one or more trades as in claim 97 wherein said preference rules further comprise one or more weights for representing an importance of at least one of said discrete variable, said continuous variable and said range variable.
99. (New) A grammar for a system that determines one or more trades as in claim 97 wherein said preference rules comprise at least one of the following:
- 10 (a) a first field representing an ideal value for said range variable;
- (b) a second field representing an ideal value for said continuous variable; and
- (c) a matrix representing one or more tradeoffs of said continuous variables.
100. (New) A grammar for a system that determines one or more trades as in claim 97 wherein said preference rules comprise a matrix representing one or more tradeoffs of said discrete variables.
- 15 101. (New) A grammar for a system that determines one or more trades as in claim 97 further comprising at least one aggregation rule comprising at least one of the following:
- (a) a list of one or more of said suppliers that can participate in the one or more trades with the buyer;
- 20 (b) one or more contribution type fields for specifying contribution types of said or more continuous variables; and
- (c) one or more constraints around the aggregation.
102. (New) A grammar for a system that determines one or more trades as in claim 101 wherein said contribution types comprise at least of the following: sum, average and zero.
- 25 103. (New) A grammar for a system that determines one or more trades as in claim 101 wherein said constraints around the aggregation comprise requiring that all orders arrive on the same day.
104. (New) A grammar for a system that determines one or more trades as in claim 99 wherein said one or more preferences rules further comprise:
- 30 (a) at least one mask for allowing at least one of said ideal value for said range variable, said continuous variable, and said one or more tradeoffs of said continuous variables to be dependent on values of said discrete variables.
105. (New) A grammar for a system that determines one or more trades as in claim 88 wherein said one or more match rules comprise at least one of the following:
- 35

- (a) a single supplier match rule describing at least one optimal one of said one or more trades with a single one of the suppliers; and
 - (b) an aggregate supplier match rule describing at least one optimal one of said one or more trades with an aggregation of said suppliers;
- 5 106. (New) A grammar for a system that determines one or more trades as in claim 105 wherein said single supplier match rule comprises at least one of the following:
- (a) an identifier for indicating said supplier of said trade;
 - (b) a utility for indicating a utility of said trade;
 - (c) a feasibility flag for indicating whether a feasible one of the trades with said single
10 supplier was found;
 - (d) a continuous variable field indicating a value for said continuous variable;
 - (e) a discrete variable field indicating a value for said discrete variable;
 - (f) a range variable field indicating a value for said range variable; and
 - (g) a cost factors field indicating constituent costs contributing to a total cost of own-
15 ership at said trade.
107. (New) A grammar for a system that determines one or more trades as in claim 104 wherein said aggregate supplier match rule comprises at least one of the following:
- (a) a utility field indicating a utility of said trade;
 - (b) a feasibility field indicating whether a feasible one of said trades with the aggrega-
20 tion of suppliers was found;
 - (c) a cost factors field indicating constituent costs contributing to a total cost of own-
ership at said trade; and
 - (d) a list of one or more trade parameters for said suppliers in the aggregation.
108. (New) A grammar for a system as in claim 107 wherein said list of trade parameters
25 comprise at least one of the following:
- (a) an identifier for identifying one of said suppliers in the aggregation;
 - (b) a continuous variable field indicating a value for said continuous variable;
 - (c) a discrete variable field indicating a value for said discrete variable;
 - (d) a range variable field indicating a value for said range variable.

AMENDMENT UNDER ARTICLE 19

In response to the International Search Report, please amend the above-identified application by substituting the attached sheets 28-41 containing the original claims 1-2 and new claims 3- 108 for the previous version of sheet 28 containing original claims 1-2.

Because substituting attached sheets 28-41 alters the page numbers of the sheets that follow the claims (i.e. the list of references, Abstract and appendices), Applicant also attaches sheets 42-60, corresponding to the previous version of sheets 29-47. No substantive changes were made to the previous version of sheets 29-47. Instead, their page numbers were changed from 29-47 to 42-60 respectively to accommodate the inclusion of the substitute claim sheets. Accordingly, please further amend the above-identified application by substituting the attached sheets 42-60 for the previous version of sheets 29-47.

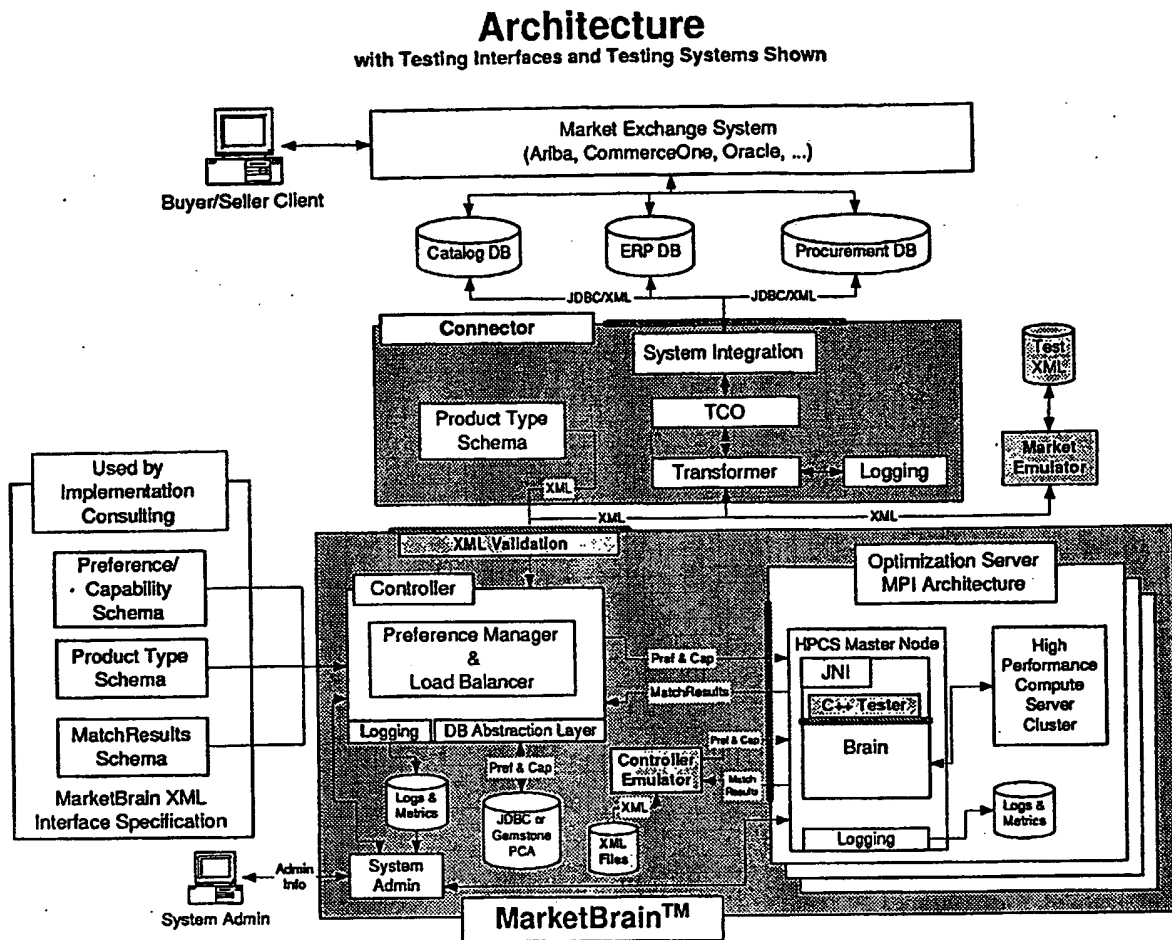


Figure 1: An architecture for the invention.

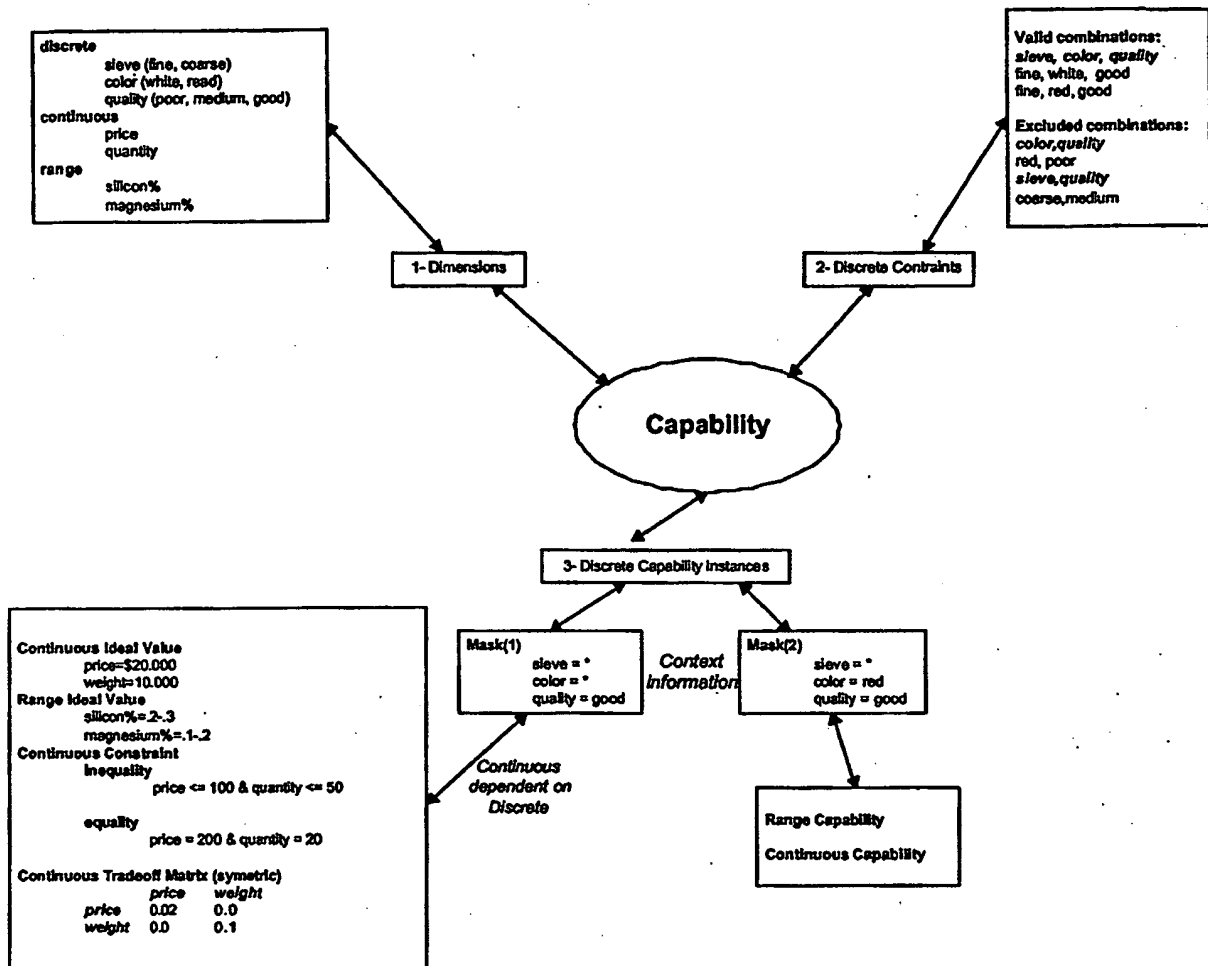


Figure 2: Schematic of a buyer-specific capability with examples indicating potential input.

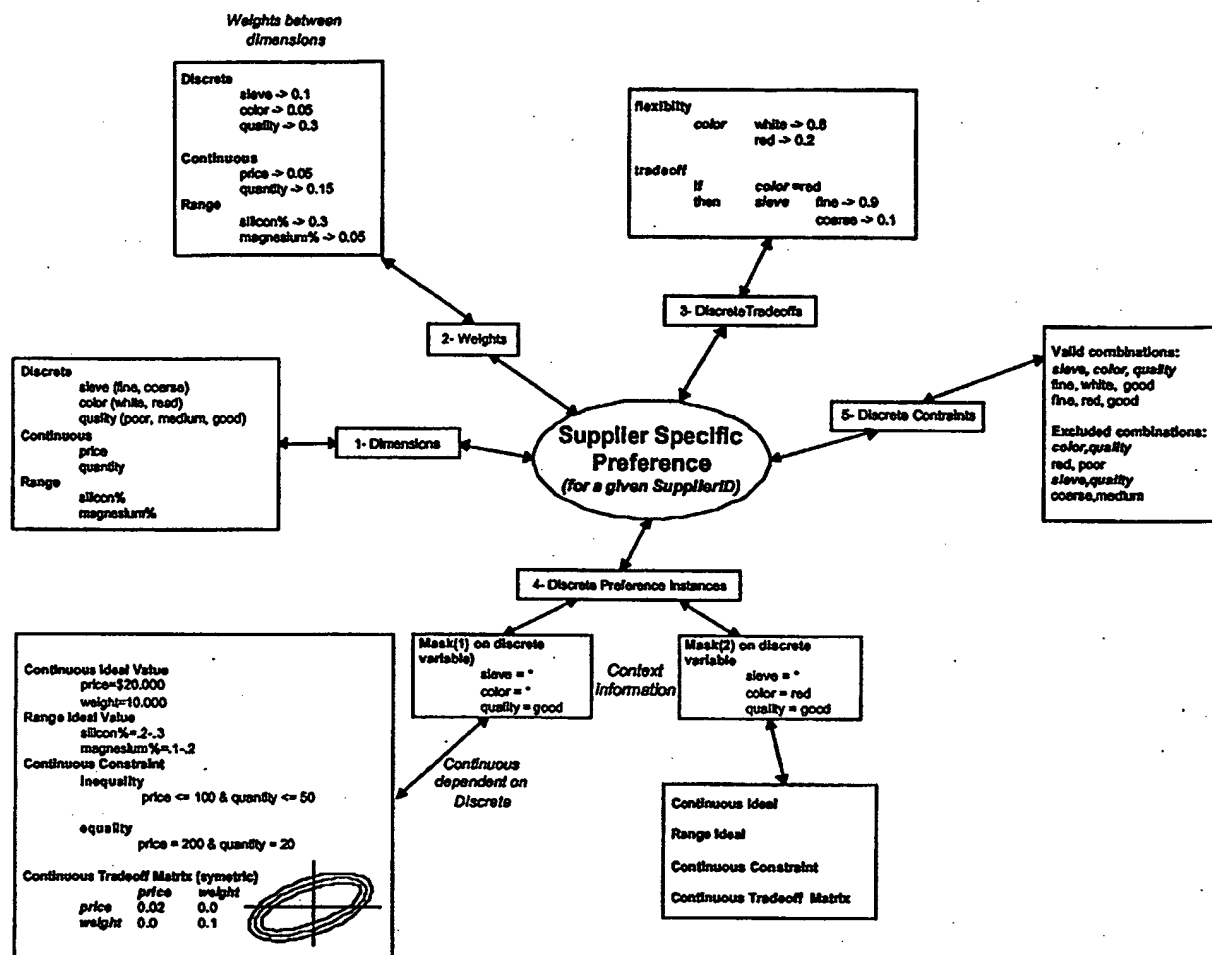


Figure 3: Schematic of a supplier-specific preference with examples indicating potential input.

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/33017**A. CLASSIFICATION OF SUBJECT MATTER**

IPC(7) : G06F 17/60

US CL : 705/37

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 705/37

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
NPLElectronic data base consulted during the international search (name of data base and, where practicable, search terms used)
PROQUEST**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y,P	LAROCHELLE, J. and SANZO B. "An Optimization Model For the Market-Mix Problem in the Banking Industry" INFOR November 2000, Vol 38 No. 4 pages 390-406	1-2
Y	SANTOS, R. "Portfolio Allocation Techniques Help Allocate Capital" Oil and Gas Journal August 9, 1999, pages 75-77	1-2

☐ Further documents are listed in the continuation of Box C.
 ☐ See patent family annex.

* Special categories of cited documents:	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A" document defining the general state of the art which is not considered to be of particular relevance	"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E" earlier document published on or after the international filing date	"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&" document member of the same patent family
"O" document referring to an oral disclosure, use, exhibition or other means	
"P" document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search 22 FEBRUARY 2001	Date of mailing of the international search report 03 APR 2001
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer VINCENT MILLIN <i>James R. Matthews</i> Telephone No. (703) 308-1065